

СОЗДАНО РАЗРАБОТЧИКАМИ ЕГЭ

ТРЕНАЖЁР

ЕГЭ

ИНФОРМАТИКА

2018

- Теория
- Тематические задания с решениями и пояснениями
 - Задания формата ЕГЭ
- Диагностический вариант экзаменационной работы
 - Контрольные варианты
 - Ответы



ЕГЭ

ТРЕНАЖЁР

С. С. Крылов
Д. М. Ушаков

ИНФОРМАТИКА

Теория

*Тематические задания
с решениями и пояснениями*

Задания формата ЕГЭ

*Диагностический вариант
экзаменационной работы*

Контрольные варианты

Ответы

*Издательство
«ЭКЗАМЕН»
МОСКВА, 2018*

УДК 372.8:002
ББК 74.263.2
К85

Крылов С. С.

К85 ЕГЭ 2018. Тренажёр. Информатика / С. С. Крылов, Д. М. Ушаков. — М. : Издательство «Экзамен», 2018. — 271, [1] с. (Серия «ЕГЭ. Тренажёр»)

ISBN 978-5-377-12371-2

Тематические тестовые задания по информатике, созданные разработчиками ЕГЭ, ориентированы на подготовку учащихся средней школы к успешной сдаче ЕГЭ.

Книга содержит множество тематических заданий для отработки каждого элемента содержания ЕГЭ по информатике, а также диагностический и контрольные варианты экзаменационной работы.

Уникальная методика подготовки, созданная разработчиками ЕГЭ, поможет учащимся научиться правильно оформлять работу, выявлять критерии оценивания, акцентировать внимание на формулировках ряда заданий и избегать ошибок, связанных с невнимательностью и рассеянностью на экзамене.

Использовать предлагаемые тестовые задания можно как в классе, так и дома. Выполнение заданий предусмотрено непосредственно в пособии.

Книга рассчитана на один учебный год, однако при необходимости позволит в кратчайшие сроки выявить пробелы в знаниях ученика и отработать задания, в которых допускается больше всего ошибок, за несколько дней до экзамена.

Издание предназначено для учителей информатики, родителей и репетиторов, а также учащихся средней школы.

Приказом № 699 Министерства образования и науки Российской Федерации учебные пособия издательства «Экзамен» допущены к использованию в общеобразовательных организациях.

УДК 372.8:002
ББК 74.263.2

Подписано в печать 25.08.2017. Формат 60х90/8. Гарнитура «Школьная».
Бумага газетная. Уч.-изд. л. 15,23. Усл. печ. л. 34.
Тираж 5000 экз. Заказ № 5038.

ISBN 978-5-377-12371-2

© Крылов С. С., Ушаков Д. М., 2018
© Издательство «ЭКЗАМЕН», 2018

СОДЕРЖАНИЕ

Введение.....	4
Диагностический вариант экзаменационной работы.....	5
<i>Ответы к диагностическому варианту экзаменационной работы</i>	18
1. Количество информации. Скорость передачи информации	24
2. Системы счисления	36
3. Кодирование информации	58
4. Основы логики	72
5. Алгоритмизация и программирование	93
6. Информационные модели.....	162
7. Определение выигрышной стратегии игры (Анализ и построение дерева игры) ...	170
8. Файловая система компьютера	185
9. Электронные таблицы	193
10. Базы данных	204
11. Сетевые технологии	213
<i>Ответы</i>	226
Контрольные варианты экзаменационной работы	232
Вариант 1.....	233
Вариант 2.....	246
<i>Ответы к контрольным вариантам экзаменационной работы</i>	260
Вариант 1.....	260
Вариант 2.....	267

ВВЕДЕНИЕ

При создании этого учебного пособия использован более чем пятилетний опыт проведения Единого государственного экзамена по информатике. В предлагаемой рабочей тетради приведены задания по всем темам экзамена, каждая глава соответствует определенной теме. Порядок расположения заданий внутри каждой главы соответствует логике изложения материала в учебниках, поэтому рабочую тетрадь можно использовать не только для обобщающего повторения перед экзаменом, но и в качестве учебного пособия в течение всего учебного года.

Поскольку данное издание представляет собой не только сборник типовых заданий в формате ЕГЭ для контроля и самоконтроля, но и учебное пособие, многие задания сформулированы как обычные задачи, в том виде, который наиболее способствует пониманию учебного материала.

В рабочей тетради не рассматриваются вопросы обучения программированию на каком-либо языке, поскольку в разных школах изучаются различные языки программирования, а рассмотрение нескольких языков не предусматривается форматом данной книги. В теме «Программирование» в качестве теоретического материала приведены общие принципы технологии программирования, соблюдение которых необходимо при выполнении заданий ЕГЭ по информатике.

Большинство заданий рабочей тетради снабжено ответами. Часть заданий намеренно оставлена авторами без ответов, для того чтобы их можно было использовать при проведении проверочных работ.

Авторы уверены, что использование данной рабочей тетради и добросовестное выполнение тренировочных заданий обязательно улучшат ваши результаты на экзамене.

Желаем успеха!

ДИАГНОСТИЧЕСКИЙ ВАРИАНТ ЭКЗАМЕНАЦИОННОЙ РАБОТЫ

Инструкция по выполнению работы

Экзаменационная работа состоит из двух частей, включающих в себя 27 заданий. Часть 1 содержит 23 задания с кратким ответом. Часть 2 содержит 4 задания с развёрнутым ответом.

На выполнение экзаменационной работы по информатике и ИКТ отводится 3 часа 55 минут (235 минут).

Ответы к заданиям 1–23 записываются в виде числа, последовательности букв или цифр. Ответ запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.

КИМ

Бланк

Ответ: 23

1	2	3																						
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Задания 24–27 требуют развёрнутого решения. В бланке ответов № 2 укажите номер задания и запишите его полное решение.

Все бланки ЕГЭ заполняются яркими чёрными чернилами. Допускается использование гелевой, или капиллярной, или перьевой ручек.

При выполнении заданий можно пользоваться черновиком. Записи в черновике не учитываются при оценивании работы.

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

Желаем успеха!

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связей (операций):

а) отрицание (инверсия, логическое НЕ) обозначается \neg (например, $\neg A$);

б) конъюнкция (логическое умножение, логическое И) обозначается \wedge (например, $A \wedge B$) либо $\&$ (например, $A \& B$);

в) дизъюнкция (логическое сложение, логическое ИЛИ) обозначается \vee (например, $A \vee B$) либо $|$ (например $A | B$);

г) следование (импликация) обозначается \rightarrow (например, $A \rightarrow B$);

е) тождество обозначается \equiv (например, $A \equiv B$). Выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо они оба истинны, либо они оба ложны);

ф) символ 1 используется для обозначения истины (истинного высказывания); символ 0 — для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения $A \rightarrow B$ и $(\neg A) \vee B$ равносильны, а $A \vee B$ и $A \wedge B$ неравносильны (значения выражений разные, например, при $A = 1$, $B = 0$).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом, $\neg A \wedge B \vee C \wedge D$ означает то же, что и $((\neg A) \wedge B) \vee (C \wedge D)$.

Возможна запись $A \wedge B \wedge C$ вместо $(A \wedge B) \wedge C$. То же относится и к дизъюнкции: возможна запись $A \vee B \vee C$ вместо $(A \vee B) \vee C$.

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле — как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

Часть 1

Ответом к заданиям 1–23 является число, последовательность букв или цифр, которые следует записать в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов и каких-либо дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведенными образцами.

1. Сколько существует натуральных чисел x , для которых выполнено неравенство $10001001_2 < x < AE_{16}$?

В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____.

2. Логическая функция F задаётся выражением $\neg a \wedge b \wedge (c \vee \neg d)$.

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных a, b, c, d .

Перем. 1	Перем. 2	Перем. 3	Перем. 4	Функция
???	???	???	???	F
0	1	0	0	1
1	1	0	0	1
1	1	0	1	1

В ответе напишите буквы a, b, c, d в том порядке, в котором идут соответствующие им столбцы (сначала — буква, соответствующая первому столбцу; затем — буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Если бы функция была задана выражением $\neg a \vee b$, зависящим от двух переменных a и b , и был приведён фрагмент её таблицы истинности, содержащий все наборы аргументов, при которых функция F истинна,

Перем. 1	Перем. 2	Функция
???	???	F
0	0	1
1	0	1
1	1	1

тогда первому столбцу соответствовала бы переменная b , а второму столбцу — переменная a . В ответе следовало бы написать: ba .

Ответ: _____.

3. Между населёнными пунктами А, В, С, D, Е, F построены дороги, протяжённость которых приведена в таблице (если ячейка пуста — дороги нет).

	A	B	C	D	E	F
A		7	2		5	
B	7			2	2	4
C	2			5	6	
D		2	5		1	3
E	5	2	6	1		
F		4		3		

Определите длину кратчайшего пути между пунктами А и F.

Ответ: _____.

4. База данных о продажах некоторого торгового предприятия состоит из трёх связанных таблиц. Ниже даны фрагменты этих таблиц.

Таблица сотрудников предприятия:

Табельный №	Фамилия	Должность	№ отдела
101	Иванов	Менеджер по продажам	6
143	Михайлова	Менеджер по продажам	5
177	Степанов	Менеджер по продажам	3
215	Петренко	Менеджер по продажам	5
216	Зеленчук	Менеджер по продажам	6
334	Зверева	Менеджер по продажам	5

Таблица заказчиков:

ID заказчика	Наименование предприятия	Город
C001	«Комета»	Москва
C002	«Метеор»	Уфа
C003	«Ракета»	Казань
C004	«Венера»	Казань
C005	«Марс»	Москва
C006	«Кратер»	Москва

Таблица продаж:

Табельный номер менеджера	ID заказчика	№ договора	Сумма договора (тыс. руб.)
215	C004	034-545	150
143	C003	324-454	200
177	C004	126-982	650
215	C006	111-456	320
216	C002	113-890	86
215	C001	129-876	111
143	C004	987-989	237

На какую сумму (в тысячах рублей) были заключены договоры о поставках в Казань менеджерами 5-го отдела? В ответе укажите только число.

Ответ: _____.

5. По каналу связи передаются сообщения, содержащие только 4 буквы А, В, С, D; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв А, В, С используются такие кодовые слова: А: 010, В: 1, С: 011. Укажите кратчайшее

кодовое слово для буквы D, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.

Ответ: _____ .

6. Автомат получает на вход четырехзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвертая цифры исходного числа.
2. Полученные два числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходное число: 5348. Суммы: $5 + 3 = 8$; $4 + 8 = 12$. Результат: 128.

Укажите наименьшее число, в результате обработки которого автомат выдаст число 1412.

Ответ: _____ .

7. Дан фрагмент электронной таблицы:

	A	B	C	D
1	8	3	7	20
2	16	=\$A1+C\$1	25	44
3	2	4		50

Формулу из ячейки B2 скопировали в ячейку C3 так, что числовое значение ячейки C3 стало отличаться от числового значения ячейки B2. Каково стало числовое значение ячейки C3?

Ответ: _____ .

8. Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	Паскаль
<pre>A = -5 C = 1024 WHILE A <> 0 C = C \ 2 A = A + 1 WEND</pre>	<pre>a := -5; c := 1024; while a <> 0 do begin c := c div 2; a := a + 1 end;</pre>
Си	Алгоритмический язык
<pre>a = -5; c = 1024; while(a != 0) { c = c / 2; a = a + 1; }</pre>	<pre>a := -5 c := 1024 нц пока a <> 0 c := div(c, 2) a := a + 1 кц</pre>
Python	
<pre>a = -5 c = 1024 while a != 0 : c = c / 2 a = a + 1</pre>	

Ответ: _____ .

9. Производится двухканальная (стерео) звукозапись с частотой дискретизации 4 кГц и 64-битным разрешением. Запись длится 1 минуту, ее результаты записываются в файл, сжатие данных не производится. Определите приблизительно размер получившегося файла (в Мбайтах). В качестве ответа укажите ближайшее к размеру файла целое число, кратное 2.

Ответ: _____.

10. Все 5-буквенные слова, составленные из букв К, Л, М, записаны в алфавитном порядке.

Вот начало списка:

1. ККККК
2. ККККЛ
3. ККККМ
4. КККЛК

.....

Запишите слово, которое стоит на 100-м месте от начала списка.

Ответ: _____.

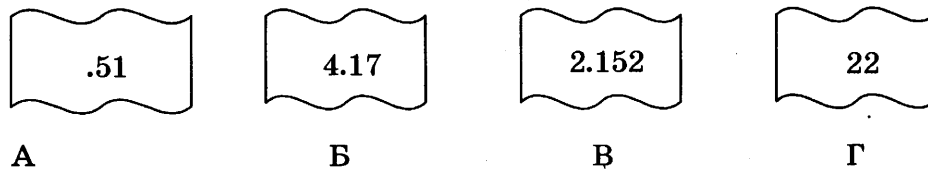
11. Ниже на пяти языках записан рекурсивный алгоритм F.

Паскаль	Си
<pre> procedure f(n:integer); begin writeln(n); if n < 7 then begin f(n+2); f(n+3) end end end </pre>	<pre> #include <stdio.h> void f(int n) { printf("%d",n); if(n<7) { f(n+2); f(n+3); } } </pre>
Бейсик	Алгоритмический язык
<pre> FUNCTION F(N) PRINT N IF N < 7 THEN F(N+2) F(N+3) END IF END FUNCTION </pre>	<pre> алг <u>цел</u> F(<u>цел</u> n) <u>нач</u> <u>вывод</u> n <u>если</u> n < 7 <u>то</u> F(n+2) F(n+3) <u>все</u> <u>кон</u> </pre>
Python	
<pre> Def F(n): print(n) if n < 7 : F(n+2) F(n+3) </pre>	

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(2)?

Ответ: _____.

12. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес.
В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



Ответ: _____.

13. В некоторой стране автомобильный номер состоит из 7 символов. Каждый символ может быть одной из 18 различных букв или десятичной цифрой.
Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байт, при этом используют посимвольное кодирование и каждый символ кодируется одинаковым и минимально возможным количеством бит.
Определите объем памяти в байтах, отводимый этой программой для записи 50 номеров. В ответе укажите только число.

Ответ: _____.

14. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
--------------------	-------------------	-------------------	--------------------

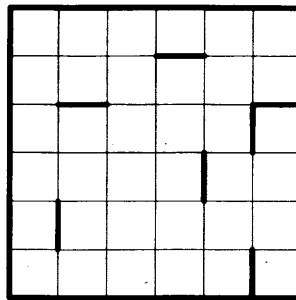
Цикл: **ПОКА** < условие > команда выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если робот начнет движение в сторону стены (находясь непосредственно рядом с ней), то он сломается и выполняемая им программа прервется.

Сколько клеток приведенного лабиринта соответствуют требованию, что, выполнив предложенную ниже программу (и не разрушившись), РОБОТ остановится в той же клетке, с которой он начал движение?

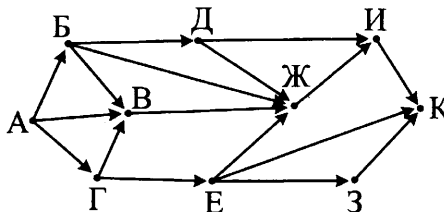
НАЧАЛО
ПОКА снизу свободно
вниз

ПОКА справа свободно
 вправо
 ПОКА сверху свободно
 вверх
 ПОКА слева свободно
 влево
 КОНЕЦ



Ответ: _____.

15. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Ответ: _____.

16. Сколько единиц содержится в двоичной записи значения выражения:

$$4^{2015} + 2^{2016} - 1?$$

Ответ: _____.

17. В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Количество найденных страниц (в тысячах)
Паровоз	18
Пароход	36
Паровоз Пароход	40

Какое количество страниц (в тысячах) будет найдено по запросу *Паровоз & Пароход* ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

18. Сколько положительных целых чисел X удовлетворяют логическому условию

$$(X \text{ нечетное} \rightarrow \text{все цифры } X \text{ различны}) \wedge X \text{ двузначное?}$$

Ответ: _____.

19. В программе описан одномерный целочисленный массив A с индексами от 1 до 10 и целочисленная переменная i . Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i = 1 TO 10 A(i) = i NEXT i FOR i = 9 TO 1 STEP -1 A(i+1) = A(10-i)+2</pre>	<pre>for i:=1 to 10 do A[i]:=i; for i:=9 downto 1 do A[i+1]:= A[10-i]+2;</pre>
Си	Алгоритмический
<pre>for (i = 1; i <= 10; i++) A[i] = i; for (i = 9; i >= 1; i--) A[i+1]= A[10-i]+2 ;</pre>	<pre>нц для i от 1 до 10 A[i]:=i кц нц для i от 9 до 1 шаг -1 A[i+1]:= A[10-i]+2 кц</pre>
Python	
<pre>for i in range(1, 11): A[i]=i for i in range(10, 0, -1): A[i+1]= A[10-i]+2</pre>	

Какое значение примет элемент массива с индексом 4 после выполнения этого фрагмента программы?

Ответ: _____.

20. Получив на вход число x , алгоритм печатает два числа L и M . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 15, а потом 3.

Паскаль	Си
<pre>var x, L, M: integer; begin readln(x); L := 0; M := 0; while x>0 do begin M := M+1; L := L + x mod 10; x := x div 10; end; writeln(L); write(M); end.</pre>	<pre>#include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L = 0; M = 0; while (x > 0) { M = M + 1; L = L + x % 10; x = x / 10; } printf("%d\n%d", L, M); }</pre>

Бейсик	Алгоритмический язык
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X > 0 M = M + 1 L = L + X MOD 10 X = X \ 10 WEND PRINT L PRINT M </pre>	<pre> алг нач цел x, L, M ввод x L := 0 M := 0 нц пока x > 0 M := M + 1 L := L + mod(x,10) x := div(x,10) кц вывод L, нс, M кон </pre>
Python	
<pre> x = int(input()) L = 0 M = 0 while x>0 do : M = M+1 L = L + x % 10 x = x // 10 print(L) print(M) </pre>	

Ответ: _____ .

21. Ниже на пяти языках приведена программа. Определите, при скольких различных значениях k программа печатает тот же результат, что и при $k = 100$ (включая $k = 100$).

Паскаль	Си
<pre> function f(n:integer):integer; var i,s : integer; begin s := 1; for i:=1 to n do s := s * 2; f := s end; var k,i : integer; begin readln(k); i:=12; while f(i)>k do i:=i-1; write(i); end. </pre>	<pre> int f(int n) { int i,s; s = 1; for(i=1 ; i<=n ; i++) s = s * 2; return s; } void main() { int k, i; scanf("%d",&k); i = 12; while (f(i)>k) i--; printf("%d", i); } </pre>

Бейсик	Алгоритмический язык
<pre> DIM K, I AS INTEGER INPUT K I = 12 WHILE F(I) > K I = I - 1 WEND PRINT I FUNCTION F(N) DIM I, S AS INTEGER S = 1 FOR I = 1 TO N S = S * 2 NEXT I F = S END FUNCTION </pre>	<pre> алг нач цел k, i ввод k i := 12 нц пока F(i) > k i := i - 1 кц вывод i кон алг цел F(цел n) нач цел s, i s := 1 нц для i от 1 до n s := s * 2 кц знач := s кон </pre>
Python	
<pre> def f(n): s = 1 for i in range(1, n+1): s = s * 2 return s k = int(input()) i = 12 while f(i)>k : i-- print(i) </pre>	

Ответ: _____ .

22. У исполнителя Калькулятор две команды:

1. Прибавь 2,
2. Умножь на 3.

Первая из них увеличивает число на экране на 3, вторая — удваивает его.

Программа для Калькулятора — это последовательность команд. Сколько есть программ, которые число 2 преобразуют в число 42?

Ответ: _____ .

23. Сколько различных решений имеет уравнение

$$(A \rightarrow B) \vee (C \rightarrow \neg D) \vee \neg (E \vee A \vee C) = 1,$$

где A, B, C, D, E — логические переменные?

В ответе не нужно перечислять все различные наборы значений A, B, C, D и E, при которых выполнено данное равенство. В качестве ответа нужно указать количество таких наборов.

Ответ: _____ .

Часть 2

Для записи ответов на задания 24–27 используйте бланк ответов № 2. Запишите сначала номер задания (24, 25 и т.д.), а затем полное решение. Ответы записывайте четко и разборчиво.

24. Требовалось написать программу, которая считывает с клавиатуры натуральное число, которое не больше 10^9 , и выводит на экран сумму цифр, которые не кратны 5. Если таких цифр нет, программа должна вывести на экран 0. Программист торопился и написал программу неправильно.

Паскаль	Си
<pre>var x, sum, k : integer; begin readln(x); sum := 0; while x >= 10 do begin k := x mod 10; if k mod 5 <> 0 then sum := sum + k; x := x div 10; end; writeln(k); end.</pre>	<pre>#include <stdio.h> void main(void) { int x, sum, k; scanf("%d", &x); sum = 0; while (x >= 10) { k = x % 10; if (k % 5 != 0) sum = sum + k; x = x / 10; } printf("%d\n", k); }</pre>
Бейсик	Алгоритмический язык
<pre>INPUT x sum = 0 WHILE x >= 10 k = x MOD 5 IF k MOD 5 <> 0 THEN sum = sum + k END IF x = x \ 10 WEND PRINT k</pre>	<pre>алг нач цел x, sum, k ВВОД x sum := 0 нц пока x >= 10 k := mod(x, 10) если mod(k, 5) <> 0 то sum := sum + k все x := div(x, 10) кц ВЫВОД k кон</pre>
Python	
<pre>x = int(input()) sum = 0 while x >= 10 : k = x % 10 if k % 5 != 0 : sum = sum + k x = x // 10 print(k)</pre>	

Последовательно выполните следующее.

1. Укажите, что выведет на экран программа при вводе числа 1234.
2. Приведите пример такого числа, при котором программа работает верно.
3. Исправьте все ошибки в программе. Для этого для каждой ошибки приведите строку, которая написана неверно, и строку, на которую ее нужно заменить, чтобы программа работала верно.

25. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 1000 — баллы учащихся выпускного класса за итоговый тест по информатике. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет найти среднее арифметическое элементов массива, делящихся нацело на 3. Известно, что в исходном массиве хотя бы один элемент делится нацело на 3.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Бейсик	Паскаль
<pre>N=30 DIM A(N) AS INTEGER DIM I, X, Y AS INTEGER DIM S AS SINGLE FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N=30; var a: array [1..N] of integer; i, x, y: integer; s: real; begin for i:=1 to N do readln(a[i]); ... end.</pre>
Си	Естественный язык
<pre>#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, x, y; float s; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<p>Объявляем массив А из 30 элементов. Объявляем целочисленные переменные I, X, Y. Объявляем вещественную переменную S. В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й. ...</p>
Python	
<pre>N = 20 i = None j = None k = None a = [int(input()) for i in range(N)] ...</pre>	

В качестве ответа необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Можно записать решение также на другом языке программирования (укажите название и ис-

пользуемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

26. Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу два камня или увеличить количество камней в куче в три раза. Например, имея кучу из 2 камней, за один ход можно получить кучу из 4 или 6 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 30. Если при этом в куче оказалось не более 45 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 16 камней и Паша утроит количество камней в куче, то игра закончится и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 29$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания.

1. а) При каких значениях числа S Паша может выиграть в один ход?

Укажите все такие значения и соответствующие ходы Паши.

- б) У кого из игроков есть выигрышная стратегия при $S = 27, 26, 24$?

Опишите выигрышные стратегии для этих случаев.

2. У кого из игроков есть выигрышная стратегия при $S = 8$? Опишите соответствующие выигрышные стратегии.

3. У кого из игроков есть выигрышная стратегия при $S = 6$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

27. На вход программе подается набор символов, заканчивающийся символом 0 (в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введен ноль, или считывать данные из файла). Ноль в этом наборе единственный. Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет составлять из всех имеющихся цифр (кроме завершающего последовательность нуля) минимальное число, состоящее ровно из трех повторяющихся одинаковых групп цифр ненулевой длины. Составленное число следует вывести на экран или в файл.

В случае невозможности составить такое число программа должна вывести «NO».

Например, пусть на вход подаются следующие символы:

fd7s22hg 547h2j 47x5 540

В данном случае программа должна вывести:

245724572457

ОТВЕТЫ К ДИАГНОСТИЧЕСКОМУ ВАРИАНТУ ЭКЗАМЕНАЦИОННОЙ РАБОТЫ

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
1	36	11	56	21	64
2	cbad	12	ГБВА	22	15
3	8	13	250	23	30
4	587	14	2		
5	00	15	9		
6	3959	16	2017		
7	36	17	14		
8	32	18	85		
9	4	19	8		
10	ЛКМКК	20	159		

24

Содержание верного ответа и указания по оцениванию
(допускаются иные формулировки ответа, не искажающие его смысла)

<p>Решение использует запись программы на Паскале. Допускается использование программы на трех других языках.</p> <p>1. Программа выведет число: 2.</p> <p>2. Пример числа, для которого программа работает правильно: 1513.</p> <p>Замечание для проверяющего. Вместо суммы цифр, не кратных 5-ти, программа выводит на экран вторую по счету. Значит, верным будет любое число, у которого сумма цифр, не кратных 5-ти, равна второй по счету цифре.</p> <p>3. В программе есть две ошибки.</p> <p>Первая ошибка: неверное условие окончания цикла.</p> <p>Строка с ошибкой: while x >= 10 do</p> <p>Верное исправление: while x > 0 do</p> <p>Вторая ошибка: неверный вывод на экран.</p> <p>Строка с ошибкой: writeln(k)</p> <p>Верное исправление: writeln(sum)</p>	
Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <p>1) указать, что выведет программа при конкретной входной последовательности;</p> <p>2) указать пример последовательности, при которой программа работает правильно;</p> <p>3) исправить первую ошибку;</p> <p>4) исправить вторую ошибку.</p> <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p>	

Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: а) выполнены три из четырех необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырех	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Возможные варианты ответа:

Бейсик	Паскаль
<pre>X=0 Y=0 FOR I = 1 TO N IF A(I)\ 3 = 0 THEN X=X+1 Y=Y+A(I) ENDIF NEXT I S = Y/X PRINT S</pre>	<pre>x:=0; y:=0; for i:=1 to N do if (A[i] mod 3 = 0) then begin x:=x+1; y:=y+A[i] end; end; s:=y/x; writeln(s);</pre>
Си	Естественный язык
<pre>x=0; y=0; for (i=0; i<N; i++) if (A[i]%3 == 0) { x=x+1; y=y+A[i]; } s=(float)y/x; printf ("%f", s);</pre>	<p>Записываем в переменные X и Y нулевые значения. В цикле от первого элемента до тридцатого проверяем делимость элементов массива на 3. Если остаток от деления текущего элемента на 3 равен 0, то увеличиваем значение переменной X на 1, а значение переменной Y — на значение этого элемента. Переходим к следующему элементу массива.</p> <p>После завершения цикла вычисляем значение среднего арифметического, деля Y на X и записывая результат в S. Выводим значение переменной S.</p>
Python	
<pre>x=0 y=0 for i in range(1,N): if A[i]%3 == 0 : x=x+1 y=y+A[i] s=y/x print(s)</pre>	

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Паша может выиграть в один ход, если $S = 28, 29$ или $S = 10, 11, 12, 13, 14, 15$. При $S = 28$ или 29 первым ходом нужно добавить в кучу два камня, при остальных указанных значениях S нужно утроить количество камней.

б) При $S = 27, 26$ или 24 утраивать количество камней не имеет смысла, так как после такого хода выигрывает противник. Поэтому можно считать, что единственный возможный ход – это добавление в кучу двух камней.

При $S = 27$ после такого хода Паши в куче станет 29 камней. В этой позиции ходящий (т.е. Валя) выигрывает (см. п. а)). Выигрышная стратегия есть у Вали.

При $S = 26$, после того как Паша своим первым ходом добавит два камня, в куче станет 28 камней. В этой позиции ходящий (т.е. Валя) выигрывает (см. п.1а)). Выигрышная стратегия есть у Вали.

При $S = 24$ выигрышная стратегия есть у Паши. Если Паша добавляет два камня, то в куче становится 26 камней. Как мы уже знаем, в этой позиции игрок, который должен ходить вторым (т.е. Паша), выигрывает.

Во всех случаях выигрыш достигается тем, что при своём ходе игрок, имеющий выигрышную стратегию, должен добавить в кучу два камня.

2. При $S = 8$ выигрышная стратегия есть у Вали. Она состоит в том, что если Паша прибавит 2 камня, то утроить кучу, а если Паша утроит первым ходом кучу, то действовать как в п. 1б).

3. При $S = 6$ выигрышная стратегия есть у Паши. Ему нужно прибавить 2 камня. Случай $S = 8$ рассмотрен в п. 2.

В таблице изображено дерево возможных партий при описанной стратегии Паши. Заключительные позиций (в них выигрывает Паша) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

И.п.	Положения после очередных ходов				
	1-й ход Паши (только ход по стратегии)	1-й ход Вали (все ходы)	2-й ход Паши (только ход по стратегии)	2-й ход Вали (все ходы)	3-й ход Паши (только ход по стратегии)
6	6+2=8	8*3=24	24+2=26	26+2=28	<u>28+2=30</u>
				<u>26*3=78</u>	
		8+2=10	<u>10*3=30</u>		

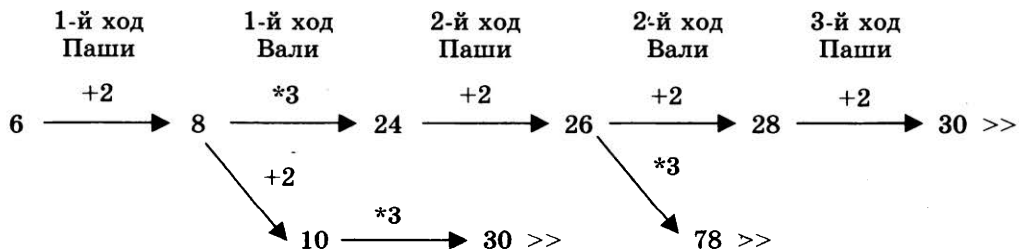


Рис.1. Дерево всех партий, возможных при Пашиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

Основные элементы правильного ответа:

Программа читает все входные данные только один раз, запоминая в целочисленном массиве из 9 элементов, сколько раз встречается соответствующая цифра во входных данных.

После этого проверяется, что все ненулевые элементы этого массива равны трем. Если это условие не выполняется или все элементы массива равны нулю, то выводится «NO».

В противном случае три раза подряд выводится упорядоченная по возрастанию группа цифр, состоящая из всех ненулевых цифр исходной строки.

Пример возможного решения (на C++):

```
#include <iostream>
using namespace std;

int main ()
{
    int num[10]; //из соображений наглядности элемент num[0] не задействован
    char c;
    for (int i=1; i<=9; i++) num[i] = 0; //инициализация
    cin >> c;
    while (c != '0')
    {
        if ((c>='1')&& (c<='9')) num[c-'0']++; //подсчет цифр
        cin >> c;
    }
    bool yes =false;
    for (int i=1; i<=9; i++)
    //проверка, есть ли в строке хотя бы одна цифра
    if (num[i]> 0){
        yes = true;
        break;
    }
    if (yes)
    {
        for (int i=1; i<=9; i++)
        //проверка, кратно ли трем количество вхождений каждой встреченной цифры
        if ((num[i]> 0) && (num[i]%3!= 0)){
            yes = false;
            break;
        }
    }
    if (yes) //вывод результата
    {
        for (int j=1; j<=3; j++)
            for (int i=1; i<=9; i++)
                for (int k=1; k<= num[i] / 3; k++) cout << i;
    }
    else cout << "NO";
    cout << '\n';
    return 0;
}
```

Пример возможного решения (на классическом Паскале):

```
program c4 (input, output);
var  num: array ['1'.. '9'] of integer;
     i, c: char;
     j, k: integer;
```

```

    yes: boolean;
begin
for i:= '1' to '9' do num[i] := 0; {инициализация}
read (c);
while c <> '0' do
begin
    if (c>='1') and (c<='9') then num[c] := num[c]+1; {подсчет цифр}
    read (c)
end;
yes := false;
i:= '1';
while (not yes) and (i<='9') do
begin
    {проверка, есть ли в строке хотя бы одна цифра}
    if num[i] > 0 then yes := true;
    i:= succ(i)
end;
if yes then
begin
    i:= '1';
    while yes and (i<='9') do
{проверка, кратно ли трем количество вхождений каждой встреченной цифры}
    begin
        if (num[i]>0) and (num[i] mod 3 <>0) then yes := false;
        i:= succ(i)
    end
end;
if yes then {вывод результата}
begin
    for j:=1 to 3 do
        for i:= '1' to '9' do
            for k:=1 to num[i] div 3 do write(i)
        end
    else write ('NO');
    writeln;
end.

```

Пример возможного решения (на Бейсике, компилятор FreeBASIC):

```

Dim num(9) As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim yes As Integer
Dim c As String
For i=1 To 9
    Num (i) = 0 'инициализация
Next i
Input c
Do While c<>"0"
    'подсчет цифр
    if (c>"0") And (c<="9") Then Num(Asc(c)-Asc("0")) = Num(Asc(c)- Asc("0"))+1
    Input c
Loop

yes = 0
For i=1 To 9
'проверка, есть ли в строке хотя бы одна цифра
    If num(i) > 0 Then
        yes = 1
    End If
Next i

```

```

Exit For
End If
Next i

If yes = 1 Then
  For i=1 To 9
    'проверка, кратно ли трем количество вхождений каждой встреченной цифры
    If (num(i)>0) And (num(i) Mod 3<> 0) Then
      yes = 0
      Exit For
    End If
  Next i
End If
c=""
If yes = 1 Then 'вывод результата
  For j=1 To 3
    For i=1 To 9
      For k=1 To num(i)/3
        c=c+Chr$(i+Asc("0"))
      Next k
    Next i
  Next j
  Print c
Else
  Print "NO"
End If

```


1. КОЛИЧЕСТВО ИНФОРМАЦИИ. СКОРОСТЬ ПЕРЕДАЧИ ИНФОРМАЦИИ

Основные понятия

Информация — первичное понятие, не имеющее общепринятого, классического определения. Одно из возможных определений: сведения об окружающем нас мире.

Для измерения информации используются специальные единицы: бит, байт, килобайт, мегабайт, гигабайт и т.д.

Минимальной (и основной) единицей измерения информации является один бит.

Бит — количество информации, которое уменьшает неопределенность в два раза.

Другое определение:

Бит — количество информации, необходимое для передачи сообщения «Да»/«Нет».

Бит может принимать только два возможных значения. Обычно их обозначают «1» и «0».

Восемь бит составляют один байт.

Для обозначения больших единиц измерения информации используются префиксы кило-, мега-, гига- и другие, например килобайт, мегабайт, гигабайт. Однако в отличие от общепризнанных значений префиксов («кило-» = 10^3 , «мега-» = 10^6 , «гига-» = 10^9) при использовании их применительно к битам и байтам они обозначают близкие степени «двойки»: «кило-» = 2^{10} , «мега-» = 2^{20} , «гига-» = 2^{30} . То есть

$$1024 \text{ байт} = 1 \text{ килобайт (Кбайт, кб)}$$

$$1024 \text{ килобайт} = 1 \text{ мегабайт (Мбайт, Мб)}$$

$$1024 \text{ мегабайт} = 1 \text{ гигабайт (Гбайт, Гб)}$$

В действительности под терминами «килобайт», «мегабайт», «гигабайт» правильнее понимать именно 10^3 , 10^6 , 10^9 . Но в ЕГЭ вместо этих терминов используются обозначения Кбайт, Мбайт, Гбайт, которые (для правильного решения задач ЕГЭ) следует принимать именно как 2^{10} , 2^{20} , 2^{30} байт.

Для вычисления количества информации применяют несколько различных способов, в зависимости от ситуации.

- Для вычисления количества информации в сообщении об одном из равновероятных событий, общее количество которых равно N , используйте формулу:

$$2^i = N,$$

где i — количество информации в сообщении.

- Для вычисления количества информации в сообщении об одном из неравновероятных событий, вероятность которого равна p , используйте формулу:

$$i = -[\log_2 p],$$

где i — количество информации в сообщении, квадратные скобки обозначают ближайшее целое, меньшее или равное значению выражения в скобках.

- Для вычисления количества информации в сообщении из k символов некоторого алфавита, в котором N различных знаков, используйте формулу:

$$I = k \cdot i,$$

где I — количество информации в сообщении, i — найти из формулы $2^i = N$.

Скорость передачи информации измеряется в битах в секунду и вычисляется по формуле:

$$V = I/t,$$

где V — скорость передачи информации, I — количество информации в сообщении, t — время передачи сообщения.

Если необходимо закодировать количество сообщений, не являющееся целой степенью числа k (различных сигналов), нужно взять ближайшее целое сверху (то есть такое наименьшее целое s , чтобы выполнялось: $k^s \geq N$).

1.26. Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 18 различных сигналов?

Ответ:																			

Рекомендация. Здесь количество различных используемых сигналов — количество состояний лампочки — 3, количество сообщений — 18.

1.27. Азбука Морзе позволяет кодировать символы для радиосвязи, задавая комбинацию точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т.д.) можно закодировать, используя код Морзе длиной не менее пяти и не более шести сигналов (точек и тире)?

Ответ:																			

Рекомендация. Так как сообщения длиной 5 и длиной 6 сигналов можно кодировать независимо одни от других (они отличаются длиной), нужно по отдельности посчитать число различных 5-символьных и 6-символьных сообщений и сложить результаты.

1.28. Вася и Петя передают друг другу сообщения, используя синий, красный и зеленый фонарики. Это они делают, включая по одному фонарику на одинаково короткое время в некоторой последовательности. Количество вспышек в одном сообщении — 3 или 4. Между сообщениями — паузы. Сколько различных сообщений могут передавать мальчики?

Ответ:																			

1.29. Для кодирования 300 различных сообщений используют 5 последовательных цветковых вспышек. Вспышки одинаковой длительности, для каждой вспышки используется одна лампочка определенного цвета. Лампочки скольких различных цветов должны использоваться при передаче (минимально возможное количество)?

Ответ:																			

1.30. В некоторой организации решили присвоить каждому сотруднику уникальный код, состоящий из символов латинского алфавита (используются все 26 заглавных букв) и цифр. При этом сначала записываются 3 латинских символа, а потом — 5 цифр. Код

- 1.39.** Вася составляет 4-буквенные слова, в которых есть только буквы В, А, С, Я, причем буква Я используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

<i>Ответ:</i>																				

- 1.40.** Вася составляет 4-буквенные слова, в которых есть только буквы Г, В, О, З, Д, Ё, причем буква Г используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

<i>Ответ:</i>																				

- 1.41.** При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: А, В, С, D, E, F, G, H, K, L, M, N. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля для каждого пользователя в системе хранятся дополнительные сведения, для чего отведено 12 байт на одного пользователя. Определите объем памяти (в байтах), необходимый для хранения сведений о 50 пользователях. В ответе запишите только целое число — количество байт.

Рекомендация. Найдем количество информации в одном символе. Так как всего возможно 12 символов, по формуле Хартли имеем $2^i \geq 12 \Rightarrow i = 4$ бита. Всего в пароле 15 символов. По формуле $I = k * i = 15 * 4 = 60$ бит. Так как для каждого пользователя хранится дополнительно 12 байт, всего получаем 60 бит + 12 байт. Так как в базе для пользователя отведено целое количество байт, переводим биты в байты. $60/8 = 7,5$. Это помещается в 8 байт. Итого $8 + 12 = 20$ байт на одного пользователя. Всего 50 пользователей. Значит, $20 * 50 = 1000$ байт.

- 1.42.** При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 13 символов и содержащий только символы из 10-символьного набора: А, В, С, D, E, F, G, H, K, L. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего отведено 8 байт на одного пользователя. Определите объем памяти (в байтах), необходимый для хранения сведений о 40 пользователях. В ответе запишите только целое число — количество байт.

<i>Ответ:</i>																				

2. СИСТЕМЫ СЧИСЛЕНИЯ

Основные понятия

Система счисления (ССч) — набор знаков, используемых для записи чисел и правила записи чисел. Эти знаки называются *цифрами*.

Набор этих цифр называется *алфавитом системы счисления*.

Количество цифр в алфавите называется *мощностью алфавита*.

Различают позиционные и непозиционные системы счисления.

Если для каждого числа системы счисления выполняется правило: вес цифры (ее значение) зависит от положения цифры в числе, такая система счисления называется *позиционной*. Если хотя бы для одного числа это правило не выполняется, система счисления называется *непозиционной*.

Пример непозиционной системы счисления — римская. В ней для числа II вес каждой цифры одинаков (равен единице).

Количество цифр в позиционной системе счисления называется *основанием* системы счисления. Именно во столько раз вес каждого разряда больше веса соседнего.

Основание — основная характеристика позиционной системы счисления.

Система счисления, которой мы пользуемся в повседневной жизни и которую изучаем в школе, — десятичная позиционная. Десятичная — потому что в ней используется десять цифр для записи чисел (от «0» до «9»), и именно в десять раз вес каждого разряда отличается от соседнего (вес сотен в десять раз больше веса десятков).

Самое маленькое основание позиционной системы счисления — 2.

Это самая простая система счисления для записи чисел, в ней всего два знака — «0» и «1». Поэтому именно двоичная система счисления используется для хранения чисел в компьютере.

Если нужно записывать числа в системе счисления, основание которой больше 10, привычных арабских цифр (от 0 до 9) не хватает и принято использовать буквы латинского алфавита: десять — A, одиннадцать — B и т.д.

Обычно используется 16-теричная система счисления.

При записи чисел в различных системах счисления принято записывать основание системы счисления справа внизу возле числа. Например, число 6 в восьмеричной системе счисления записывают: 6_8 . Если основание системы счисления справа внизу возле числа не указано, считается, что это десятичная система счисления.

Для перевода числа из какой-либо системы счисления в десятичную необходимо:

- 1) пронумеровать разряды числа справа налево, начиная с нуля;
- 2) умножить каждую цифру числа на основание его системы счисления, возведенное в степень номера этого разряда;
- 3) сложить полученные числа.

Для перевода десятичного числа в другую систему счисления необходимо:

- 1) делить нацело с остатком число на нужное основание системы счисления;
- 2) получившееся частное (целое) тоже делить нацело с остатком на это основание;
- 3) продолжать деления до тех пор, пока частное не получится равно нулю;
- 4) выписать остатки в порядке, обратном их получению.

Практическая часть

2.1. Расставьте необходимые термины напротив их определений.

Термины: система счисления, алфавит системы счисления, мощность алфавита, основание системы счисления.

- _____ а) количество цифр, используемых при записи чисел
_____ б) набор цифр, используемых при записи чисел, и правила записи чисел
_____ в) правила записи цифр
_____ г) набор цифр, используемых при записи чисел
_____ д) количество цифр в алфавите позиционной системы счисления

2.2. Чем отличается позиционная система счисления от непозиционной?

Ответ: _____

2.3. В позиционной системе счисления во сколько раз вес (значение) каждого разряда больше предыдущего?

Ответ: _____

2.4. В N -ичной системе счисления для записи чисел используется _____ различных цифр. Самая маленькая цифра равна _____. Самая большая цифра равна _____.

2.5. В N -ичной системе счисления число, которое на 1 больше, чем самая старшая цифра, записывается как _____.

2.6. В двоичной системе счисления для записи чисел используются только _____ и _____.

2.7. В шестнадцатеричной системе счисления кроме обычных десяти арабских цифр (от 0 до 9) используются также _____ букв латинского алфавита: от _____ до _____. Цифра «десять» записывается как _____, цифра «_____» записывается как F .

2.8. Обозначьте на рисунке следующие термины (обведите и подпишите или напишите термин и стрелками укажите их): разряд, номер разряда, основание системы счисления.

4 3 2 1 0
5 2 4 3 6 8

2.9. В шестнадцатеричной системе счисления между числами $2B_{16}$ и $2E_{16}$ находятся числа _____₁₆ и _____₁₆.

2.10. Заполните пустые клетки таблицы последовательными числами в системах счисления с основанием 3, 4, 5 (таблицу соответствия между десятичной, двоичной и шестнадцатеричной системами счисления мы рекомендуем вам выучить наизусть).

Система счисления

10	2	8	16	3	4	5
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	10	2	2			
3	11	3	3			
4	100	4	4			
5	101	5	5			
6	110	6	6			
7	111	7	7			
8	1000	10	8			
9	1001	11	9			
10	1010	12	A			
11	1011	13	B			
12	1100	14	C			
13	1101	15	D			
14	1110	16	E			
15	1111	17	F			
16	10000	20	10			
17	10001	21	11			

2.11. После числа: 100111_2 212_3 37_8 BF_{16} 21333_4 66_7
 следует число: _____₂ _____₃ _____₈ _____₁₆ _____₄ _____₇

2.12. Числу: 10100_2 2100_3 520_8 $A00_{16}$ 3120_4 50_7
 предшествует
 число: _____₂ _____₃ _____₈ _____₁₆ _____₄ _____₇

2.13. Между числами: 1111_2 и 10001_2 BF_{16} и $C1_{16}$ 2221_3 и 10000_3 109_{16} и $10B_{16}$
 стоит число: _____₂ _____₁₆ _____₃ _____₁₆

2.14. В каждом столбце обведите большее число и подчеркните меньшее:

10010_2	2010_3	507_8	$A00_{16}$	10000_4	$1FF_{16}$
1111_2	1112_3	277_8	EOF_{16}	30000_4	$F00_{16}$
10100_2	2212_3	374_8	$10D_{16}$	23012_4	333_{16}
11000_2	1222_3	630_8	CFF_{16}	32100_4	ABC_{16}

2.15. Расставьте цифры — порядок выполнения действий при переводе из любой системы счисления в десятичную:

- сложить все числа
- пронумеровать разряды числа справа налево, начиная с нуля
- цифру каждого разряда умножить на основание системы счисления, возведенное в степень номера разряда

2.16. Расставьте цифры — порядок выполнения действий при переводе из десятичной системы счисления в любую другую:

- выписать остатки от деления слева направо в порядке, обратном их получению
- повторять действие ___ до тех пор, пока частное от деления не будет равно нулю
- поделить число нацело с остатком на основание системы счисления, в которую переводим

Напоминание. При переводе из двоичной системы счисления в десятичную цифры, которые нужно умножать на число 2 в какой-то степени, равны 0 или 1. Все, что умножено на 0, все равно дает ноль. Поэтому эти слагаемые лучше просто опустить. Все, что умножено на 1, таким же и остается. Поэтому эти умножения тоже лучше опустить. Получается, достаточно просто сложить степени «двойки», в разрядах которых стоят «единицы».

2.17. Переведите числа в десятичную систему счисления (заполните пропущенные в клетках цифры):

$$\begin{array}{cccc} \square & \square & \square & \square \\ 2 & 3 & 1 & 4 \\ \end{array} 5 = \square \cdot 5^3 + 3 \cdot 5^{\square} + 1 \cdot \square^1 + \square \cdot 5^0 = 250 + \square\square + \square + 4 = \square\square\square_{10}$$

$$\begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ 1 & 0 & 1 & 0 & 0 & 1 \\ \end{array} 2 = 2^6 + 2^{\square} + 2^{\square} + 1 = \square\square + 16 + \square + 1 = \square\square_{10}$$

$$\begin{array}{ccc} \square & \square & \square \\ 2 & A & B \\ \end{array} 16 = \square \cdot 16^2 + 10 \cdot \square\square^{\square} + \square\square = 512 + \square\square\square + \square\square = \square\square\square_{10}$$

2.18. Переведите числа в десятичную систему счисления:

$$1010110_2 = \underline{\hspace{10cm}}$$

$$21020_3 = \underline{\hspace{10cm}}$$

$$526_8 = \underline{\hspace{10cm}}$$

$$3CE_{16} = \underline{\hspace{10cm}}$$

$$3021_4 = \underline{\hspace{10cm}}$$

$$256_7 = \underline{\hspace{10cm}}$$

Для успешного выполнения следующих заданий по системам счисления необходимо хорошо помнить материал младшей школы.

Для эффективного использования этого метода необходимо наизусть знать степени числа «2». Вообще говоря, это знание очень поможет вам решать множество задач ЕГЭ по информатике. Мы рекомендуем заранее выучить степени двойки хотя бы до 10-й, а лучше — до 16-й.

Напоминание. Две системы счисления будем называть **родственными**, если основание одной системы счисления равно степени основания другой. Например, 2 и 8, 2 и 16, 3 и 9.

Для произвольной пары систем счисления, чтобы перевести число из одной системы в другую, нужно осуществлять два перевода — сначала из исходной в десятичную, потом из десятичной в нужную. В родственных системах счисления можно осуществлять перевод напрямую. Чтобы не описывать процесс в общем виде (который вам, вероятно, никогда не понадобится), мы остановимся на системах счисления $2 \leftrightarrow 8$, $2 \leftrightarrow 16$.

Замечание. Вы будете гораздо быстрее осуществлять действия по переводу чисел в родственных системах счисления, если выучите наизусть таблицу соответствия цифр от 0 до 15 в двоичной, восьмеричной и шестнадцатеричной системах счисления (таблицу мы привели несколькими страницами ранее). Это существенно экономит вам время на экзамене.

При решении нижеследующих задач настоятельно рекомендуем этой таблицей не пользоваться — либо выучите ее наизусть, либо научитесь переводить числа «на лету». Если вы будете просто заглядывать в таблицу — не научитесь делать это сами и не сможете осуществлять перевод на экзамене.

2.25. Расставьте цифры — порядок выполнения действий при переводе из двоичной системы счисления в восьмеричную:

- записать получившиеся цифры в том же порядке, в котором записаны группы разрядов
- сгруппировать разряды группами по 3, справа налево
- двоичное число в каждой группе перевести в десятичную систему счисления

2.26. Переведите число 10001101_2 в восьмеричную систему счисления:

$$\begin{array}{r} \underline{1000} \underline{1101}_2 \\ \square \square \quad \square \end{array} \quad \begin{array}{l} 2 \quad 1 \quad 0 \\ 101_2 = 2^2 + 2^0 = 4 + 1 = 5 \end{array}$$

2.27. Переведите числа 11111100_2 и 1010110_2 в восьмеричную систему счисления:

$$\begin{array}{r} \underline{11111} \underline{100}_2 \\ \square \square \quad \square \end{array} \quad \begin{array}{r} \underline{10101} \underline{10}_2 \\ \square \square \quad \square \end{array}$$

2.28. Расставьте цифры — порядок выполнения действий при переводе из двоичной системы счисления в шестнадцатеричную:

- записать получившиеся цифры в том же порядке, в котором записаны группы разрядов
- сгруппировать разряды группами по ____, справа налево
- двоичное число в каждой группе перевести в десятичную систему счисления
- получившееся десятичное число перевести в 16-ю цифру

2.29. Переведите число 11010101101_2 в шестнадцатеричную систему счисления:

$$\begin{array}{c} \underbrace{11010101101}_2 \\ \square \quad \square \quad \square \\ \parallel \quad \parallel \quad \parallel \\ \square \quad \square \quad \square \end{array} \quad \begin{array}{c} 3210 \\ 1101_2 = 2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13 \\ \square \quad \square \quad \square \end{array} \quad \begin{array}{c} 13 \\ \square \quad \square \quad \square \\ D_{16} \end{array}$$

2.30. Переведите числа 111111100_2 и 11101011110_2 в шестнадцатеричную систему счисления:

$$\begin{array}{c} 111111100_2 \\ \square \quad \square \quad \square \\ \square \quad \square \quad \square \end{array} \quad \begin{array}{c} 11101011110_2 \\ \square \quad \square \quad \square \\ \square \quad \square \quad \square \end{array}$$

2.31. Расставьте цифры — порядок выполнения действий при переводе из восьмеричной системы счисления в двоичную:

- выписать тройки двоичных разрядов друг за другом в том же порядке, в котором стоят цифры восьмеричного числа
- каждую цифру восьмеричного числа перевести в двоичную систему счисления
- если двоичное представление цифры состоит меньше чем из трех разрядов, дописать слева один или два нуля, чтобы всего получилось ровно три разряда

Пример.

$$\begin{array}{c} 532_8 \\ \swarrow \quad \downarrow \quad \searrow \\ \underbrace{101011010}_2 \end{array} \quad \begin{array}{l} 5 = 4 + 1 = 2^2 + 2^0 = 101_2 \\ 3 = 2 + 1 = 2^1 + 2^0 = 11_2 \\ 11_2 = 011_2 \end{array}$$

2.32. Переведите число 714_8 в двоичную систему счисления:

$$\begin{array}{c} 714_8 \\ \swarrow \quad \downarrow \quad \searrow \\ \square \square 1 \quad \square \square \square \quad 1 \square \square \end{array} \quad \begin{array}{l} 7 = 4 + \square + 1 = 2^{\square} + 2^1 + 2^{\square} = \square \square 1_2 \\ 4 = 2^{\square} = 1 \square \square_2 \\ 1 = 1_2 = \square \square \square_2 \end{array}$$

2.33. Переведите числа 603_8 и 325_8 в двоичную систему счисления:

$$\begin{array}{c} 603_8 \\ \swarrow \quad \downarrow \quad \searrow \\ \square \square \square \square \square \square \square \square \end{array} \quad \begin{array}{c} 325_8 \\ \swarrow \quad \downarrow \quad \searrow \\ \square \square \square \square \square \square \square \square \end{array}$$

2.34. Расставьте цифры — порядок выполнения действий при переводе из шестнадцатеричной системы счисления в двоичную:

- выписать четверки двоичных разрядов друг за другом в том же порядке, в котором стоят цифры шестнадцатеричного числа

- если двоичное представление цифры состоит меньше чем из ___ разрядов, дописать слева нули, чтобы всего получилось ровно ___ разряда
- каждую цифру шестнадцатеричного числа перевести в двоичную систему счисления

2.35. Переведите число $D27_{16}$ в двоичную систему счисления:

$D27_{16}$

$D_{16} = 13 = 8 + 4 + 1 = 2^3 + 2^2 + 2^0 = 1101_2$

$2 = 2^{\square} = \square\square\square\square_2$

$7 = \square + 2^{\square} + 1 = 2^{\square} + 2^{\square} + 2^{\square} = \square\square\square\square_2$

2.36. Переведите числа $E43_{16}$ и $A0C_{16}$ в двоичную систему счисления:

$E43_{16}$

$A0C_{16}$

2.37. Дано: $a = D7_{16}$, $b = 331_8$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

1) 11011001 2) 11011100 3) 11010111 4) 11011000

Рекомендация. Так как все варианты ответов даны в двоичной системе счисления, имеет смысл перевести числа a и b в двоичную систему счисления, выбрать из них меньшее, прибавить к нему столбиком «1» и поискать результат среди вариантов ответа.

2.38. Дано: $a = 263_8$, $b = B5_{16}$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

1) 10110010 2) 10110101 3) 10110110 4) 10110100

2.39. Дано: $a = C6_{16}$, $b = 310_8$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

1) 11000111 2) 11000101 3) 11001001 4) 11001000

2.40. Дано: $a = A4_{16}$, $b = 246_8$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

1) 10100110 2) 10100111 3) 10100101 4) 10100011

2.41. Дано: $a = 341_8$, $b = E3_{16}$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

1) 11100000 2) 11100001 3) 11100010 4) 11100100

Сначала запишем оба исходных числа друг под другом, выровняв их по правому краю (по младшему разряду):

$$\begin{array}{r} 2465 \\ + 637 \\ \hline \end{array}$$

Начинаем с младшего разряда (самого правого). Сложим цифры. $5 + 7 = 12$. Цифру «2» запишем в этом разряде, «1» (которая на самом деле 10) — это перенос в следующий разряд:

$$\begin{array}{r} 2465 \\ + 637 \\ \hline 2 \\ \wedge \end{array}$$

В следующем разряде складываем $6 + 3 + 1$ (из предыдущего разряда) = 10.

Цифру «0» пишем в этом разряде, «1» — перенос в следующий разряд:

$$\begin{array}{r} 2465 \\ + 637 \\ \hline 02 \\ \wedge \wedge \end{array}$$

Складываем $4 + 6 + 1 = 11$. «1» пишем, «1» — перенос в следующий разряд:

$$\begin{array}{r} 2465 \\ + 637 \\ \hline 102 \\ \wedge \wedge \wedge \end{array}$$

В самом левом (старшем) разряде стоит «2», плюс «1» перенесли из предыдущего разряда. Итого «3»:

$$\begin{array}{r} 2465 \\ + 637 \\ \hline 3102 \\ \wedge \wedge \wedge \end{array}$$

Теперь попробуем сделать то же самое, но в восьмеричной системе счисления:

$$2465_8 + 637_8$$

Запишем оба исходных числа друг под другом, выровняв их по правому краю (по младшему разряду):

$$\begin{array}{r} 2465_8 \\ + 637_8 \\ \hline \end{array}$$

Начинаем с младшего разряда (самого правого). Сложим цифры. $5 + 7 = 12$. Но вот только это совсем теперь не значит, что «2» — пишем, а «1» — переносим. Потому что получившееся у нас 12 — это десятичное число. А нам нужно восьмеричное. Поэтому мы переводим 12 в восьмеричную систему счисления. В общем случае нужно бы, конечно, делить «уголком» и выписывать остатки от деления. Но при сложении двух чисел в любой системе счисления не может получиться перенос в следующий разряд больше, чем «1». Поэтому результат деления всегда будет давать «1» в качестве частного (если, конечно, случился перенос). А это значит, что вместо деления «уголком» достаточно просто вычесть из получившейся суммы основание системы счисления. В данном случае $12 - 8 = 4$. То есть «4» пишем, «1» — переносим:

$$\begin{array}{r} 2465_8 \\ + 637_8 \\ \hline 4_8 \\ \wedge \end{array}$$

В следующем разряде складываем $6 + 3 + 1$ (из предыдущего разряда) = 10. Это ≥ 8 . Значит, перенос есть. Вычитаем 8: $10 - 8 = 2$.

Цифру «2» пишем в этом разряде, «1» — перенос в следующий разряд:

$$\begin{array}{r} 2465_8 \\ + 637_8 \\ \hline 24_8 \\ \wedge \wedge \end{array}$$

Складываем $4 + 6 + 1 = 11$. Перенос есть. Вычитаем 8. «3» пишем, «1» — перенос в следующий разряд:

$$\begin{array}{r} 2465_8 \\ + 637_8 \\ \hline 324_8 \\ \wedge \wedge \wedge \end{array}$$

В самом левом (старшем) разряде стоит «2», плюс «1» перенесли из предыдущего разряда. Итого «3». Переноса нет:

$$\begin{array}{r} 2465_8 \\ + 637_8 \\ \hline 3324_8 \\ \wedge \wedge \wedge \end{array}$$

2.65. Чему равна сумма чисел 43_8 и 56_8 (в восьмеричной системе счисления)?

<i>Ответ:</i>																			

2.66. Чему равна сумма чисел 110110110_2 и 1110111_2 (в двоичной системе счисления)?

<i>Ответ:</i>																			

2.67. Чему равна сумма чисел 132_4 и 233_4 (в четверичной системе счисления)?

<i>Ответ:</i>																			

Замечание. При действиях в шестнадцатеричной системе счисления нужно помнить также о том, что если в разряд нужно записать число, например 13, — это не значит, что «3» пишем, а «1» — переносим. Это значит, что в этом месте нужно написать шестнадцатеричную цифру «D».

Пример. Найти сумму чисел DF_{16} и $B7_{16}$ в шестнадцатеричной системе счисления.

Решение.

В младшем разряде: $5 + 7 = 12$. Это не значит, что «2» — пишем, а «1» — переносим. Так как $12 \leq 16$, то переноса нет. Нужно просто вспомнить, что $12 = C_{16}$:

$$\begin{array}{r} DF_{16} \\ + B7_{16} \\ \hline C_{16} \end{array}$$

Рекомендация. Обозначим искомое основание как N . Переведем число 60_N в десятичную систему счисления. Получим: $6 \cdot N^1 + 0 \cdot N^0 = 6N$. Известно, что это равно 48. Решаем уравнение $6N = 48$ и находим N .

2.80. Число 36 в системе счисления с некоторым основанием записывается как 40. Укажите это основание.

Ответ:	
--------	--

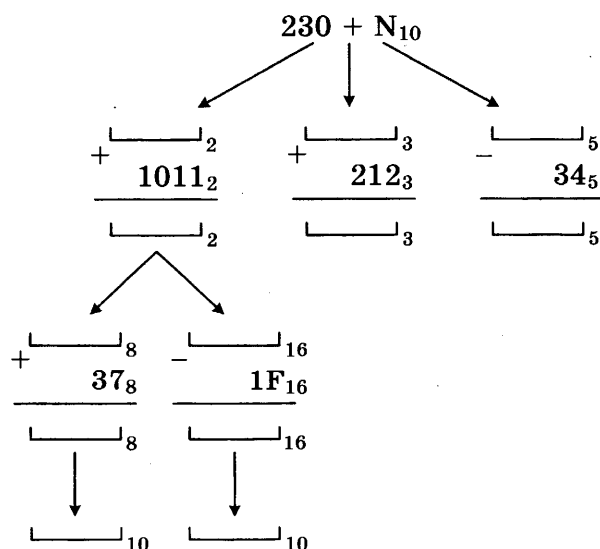
2.81. Число 72 в системе счисления с некоторым основанием записывается как 40. Укажите это основание.

Ответ:	
--------	--

2.82. Комплексное задание в системах счисления.

- 1) Получить исходное число: $230 + N$ (выберите N от 1 до 25).
- 2) Перевести число (столбиком) в двоичную, троичную, пятеричную ССч.
- 3) К полученному двоичному числу прибавить 1011_2 (столбиком).
- 4) Результат перевести в восьмеричную и шестнадцатеричную ССч (объединением разрядов).
- 5) К полученному восьмеричному числу прибавить (столбиком) 37_8 .
- 6) Результат перевести в десятичную ССч.
- 7) Из полученного шестнадцатеричного числа отнять (столбиком) $1F_{16}$.
- 8) Результат перевести в десятичную ССч.
- 9) К полученному троичному числу прибавить (столбиком) 212_3 .
- 10) Из полученного пятеричного числа отнять (столбиком) 34_5 .

Схема выполнения задания:



$$\begin{array}{r} _100000000_3 \\ \underline{100000_3} \\ 22200000_3 \end{array}$$

То есть, $3^8 - 3^5$ — это 3 двойки и 5 нулей. Нетрудно сделать вывод, что $3^x - 3^y$ — это $(x - y)$ двоек и (y) нулей. Значит, в нашем примере $3^{100} - 3^1$ — это 99 двоек и еще 0. А при добавлении к этому 3^{400} в начале троичной записи добавится еще одна единица. Добавление 3^0 повлияет только на нулевую позицию результата. Там стоит 0. Значит, в ответе на нулевой позиции будет стоять 1 и это не повлияет на остальные разряды. Значит, *ответ* — 99 двоек.

2.95. Сколько единиц содержится в двоичной записи значения выражения: $8^{10} + 2^{36} - 15$?

Ответ:																																								

2.96. Значение арифметического выражения: $9^{30} + 3^{70} - 25$ — записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Ответ:																																								

2.97. Значение арифметического выражения: $25^{30} + 4 \cdot 5^{70} - 21$ — записали в системе счисления с основанием 5. Сколько цифр «4» содержится в этой записи?

Ответ:																																								

3. КОДИРОВАНИЕ ИНФОРМАЦИИ

Основные понятия

Для удобства представления информации в компьютере все возможные виды информации переводятся в числовую форму, и эти числа хранятся в компьютере в двоичном виде.

Для кодирования текстовой информации используются специальные таблицы, в которых каждому символу (в т.ч. каждой букве) поставлен в соответствие некоторый номер. Так как эти номера хранятся в двоичном виде, принято говорить, что каждому символу соответствует свой двоичный код. Длина этих двоичных кодов одинакова для всех символов таблицы.

Основной (традиционной) таблицей кодирования символов является таблица ASCII (американский стандартный код обмена информацией, читается «АСКИ»). В ней для хранения каждого символа выделено 8 бит (1 байт). Таким образом, общее количество различных символов, которое кодируется таблицей ASCII, равно $2^8 = 256$.

Для кодирования русских букв, в зависимости от операционной системы, используется одна из разновидностей таблицы ASCII (так называемая *кодовая страница*): КОИ-8, Windows-1251, ISO, DOS, MAC.

Из-за ограниченности количества различных символов в таблице ASCII на смену ей придумана кодовая таблица Unicode (Юникод). В ней для хранения каждого символа используется 16 бит (2 байта). Таким образом, общее количество различных символов, которое кодируется таблицей Unicode, равно $2^{16} = 65\,536$.

Для кодирования информации о цвете используется таблица цветов. Количество цветов N , которое может быть закодировано при помощи i бит, вычисляется по формуле:

$$2^i = N.$$

Для кодирования полноцветных изображений используют цветовые модели (способ представления информации о цвете через несколько характеристик цвета). Одна из основных цветовых моделей — RGB. Она хранит информацию о цвете в виде яркости трех базовых цветов — красного (*red*), зеленого (*green*) и синего (*blue*), каждая из которых может принимать значения от 0 до 255. Эти три цвета «складываются» при восприятии их человеком, и он воспринимает их как некоторый оттенок цвета. Эта цветовая модель (RGB) называется *аддитивной*, потому что яркости базовых цветов в ней складываются (*add*). Чем больше суммарная яркость, тем светлее общий оттенок цвета и наоборот.

То есть, если все три яркости равны нулю, получается черный цвет, если 255, то белый. Если все три яркости одинаковы и лежат между 0 и 255, получается оттенок серого цвета. Если в коде цвета присутствует только одна яркость из трех, получается оттенок этого цвета.

Для кодирования растровых изображений цвет каждого пикселя изображения записывается своим кодом (одинаковой длины для всех пикселей изображения). Таким образом, общий объем памяти, необходимый для хранения неупакованного растрового изображения, можно вычислить, умножив количество пикселей на длину кода цвета одного пикселя. Число пикселей в растровом изображении — ширина, умноженная на высоту. Так как длина кода цвета обычно записана в битах, а объем памяти принято считать в байтах, получившуюся величину нужно поделить на 8. Получаем формулу:

$$V = H \times W \times C / 8,$$

где H и W — высота и ширина изображения в пикселях,

C — число бит, используемое для хранения кода цвета одного пикселя,

V — объем памяти для хранения этого изображения в байтах.

Кодирование звука. Нужно понимать, что с точки зрения компьютера звук — это колеблющаяся с определенной частотой мембрана (при воспроизведении звука колебания мембраны создают колебания воздуха (звуковую волну), которую слышит ухо). Для кодирования звука записывают отклонение мембраны от нулевого положения определенное количество раз в секунду. Это количество раз в секунду называют частотой дискретизации. Значение отклонения мембраны кодируют в двоичной системе счисления. Это может быть осуществлено с разной точностью. Значение максимального отклонения разбивают на какое-то количество делений (квантуют). Это количество делений называется количеством уровней квантования. Так как отклонение записывают в двоичной системе счисления, количество уровней квантования выбирают равным степени двойки. Соответственно, количество уровней квантования связано с количеством бит, требуемым для кодирования, все тем же соотношением:

$$2^i = N.$$

Здесь N — количество уровней квантования,

i — количество бит, которым кодируется каждое отклонение мембраны.

Общая формула вычисления объема звукового файла:

$$V = Ch \times v \times t \times i / 8,$$

где Ch — количество каналов (обычно 1 (моно), 2 (стерео) или 4 (квадро)),

v — частота дискретизации в Герцах,

t — время звучания/звукозаписи в секундах,

i — число бит разрешения (делим на 8, чтобы получить в ответе байты),

V — объем памяти для хранения этого звукового фрагмента в байтах.

Практическая часть

3.1. Выберите правильный ответ. В каком виде хранится вся информация в компьютере? (Обведите букву, соответствующую правильному утверждению.)

- а) Для каждого вида информации придуман свой тип файла.
- б) В двоичном (для всех видов информации придуманы свои двоичные коды).
- в) Числа хранятся как числа, буквы — как буквы, изображения — как точки.

3.2. Укажите одно или несколько правильных утверждений. Буквы, соответствующие правильным утверждениям, обведите, остальные — зачеркните.

Относительно кодирования текстовой (символьной) информации можно сказать следующее:

- а) для каждого символа выделен свой код. Эти коды хранятся в виде двоичных чисел одинаковой длины;
- б) часто для кодирования символов используется таблица, состоящая из 256 различных символов. Она называется ASCII. Для хранения каждого символа выделено 8 бит (1 байт);
- в) при кодировании русских букв в коде ASCII используется один из пяти различных способов: КОИ-8, Windows-1251, ISO, DOS, MAC. Обычно способ кодирования выбирается в зависимости от используемой операционной системы;
- г) часто для кодирования символов используется таблица, состоящая из 65 536 различных символов. Она называется Unicode. Для хранения каждого символа выделено 16 бит (2 байта);

Из четырех полученных сообщений в этой кодировке только одно прошло без ошибки и может быть корректно декодировано. Найдите его:

1) 110100000100110011

3) 110100001001100111

2) 111010000010010011

4) 110110000100110010

3.12. Через канал связи передается сообщение, состоящее только из символов А, Б, В, Г и Д. Каждый символ при этом кодируется своим собственным кодом: А–11, Б–100, В–00, Г–01. Каким самым коротким кодовым словом нужно закодировать символ Д, чтобы код удовлетворял свойству однозначного декодирования?

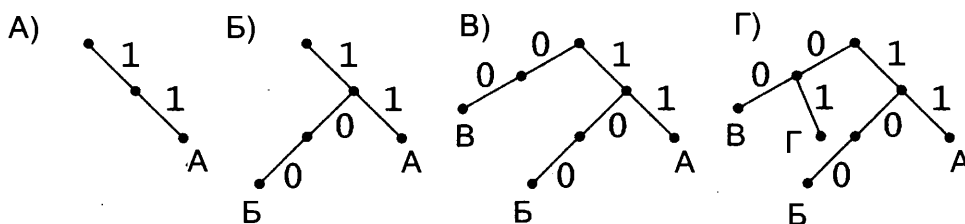
1) 1000

2) 10

3) 101

4) 0

Рекомендация. Чтобы код удовлетворял свойству однозначного декодирования, он должен быть префиксным. То есть никакое кодовое слово не должно быть началом никакого другого кодового слова. Исследуем имеющиеся кодовые слова и построим дерево декодирования. В каждом узле дерева декодирования принимается решение, в какую сторону «повернуть» — направо или налево. Будем рисовать начиная от некоего начального (корневого) узла, ветки. Для «1» будем рисовать ветку направо, для «0» — налево:



Начинаем добавлять в дерево первую букву (А). Ее код — две единицы, рисуем две ветки вправо. Потом добавляем вторую букву (Б). Ее код — 100, одна ветка вправо (уже есть), от этого узла две ветки влево. Добавляем третью букву (В). Ее код — два нуля, две ветки влево. Добавляем четвертую букву (Г). Ее код — 01, одна ветка влево (уже есть), одна ветка вправо. Для всех известных кодов букв построили дерево декодирования. Теперь ищем в получившемся дереве, у какого узла нет второй ветки. Это узел «10». У него отсутствует ветка вправо. То есть ветка «1». Получается, что у недостающей буквы самое короткое кодовое слово должно быть 101.

3.13. Через канал связи передается сообщение, состоящее только из символов А, Б, В, Г и Д. Каждый символ при этом кодируется своим собственным кодом: А–111, Б–0, В–101, Г–110. Каким самым коротким кодовым словом нужно закодировать символ Д, чтобы код удовлетворял свойству однозначного декодирования?

1) 1001

2) 1

3) 10

4) 100

3.14. Через канал связи передается сообщение, состоящее только из символов А, Б, В, Г и Д. Каждый символ при этом кодируется своим собственным кодом: А–11, Б–10, В–011, Г–010. Каким самым коротким кодовым словом нужно закодировать символ Д, чтобы код удовлетворял свойству однозначного декодирования?

1) 0

2) 00

3) 001

4) 111

3.15. Через канал связи передается сообщение, состоящее только из символов А, Б, В, Г и Д. Каждый символ при этом кодируется своим собственным кодом: А–110, Б–111, В–100, Г–101. Каким самым коротким кодовым словом нужно закодировать символ Д, чтобы код удовлетворял свойству однозначного декодирования?

1) 0

2) 00

3) 000

4) 0000

Напоминание. Для кодирования цвета используются цветовые модели.

Цветовая модель — способ сопоставления каждому используемому в модели цвету своего кодового слова.

Считается, что человеческий глаз способен различать 16 миллионов оттенков цвета.

Так как составить таблицу всех 16 миллионов цветов и назвать каждый представляется затруднительным и малоэффективным способом, придумали правила, по которым значение цвета хранится как числовое значение его определенных характеристик.

Самый распространенный такой способ — представлять цвет как интенсивность свечения трех базовых цветов, из которых можно составить любой из 16 миллионов цветов. (Вообще говоря, правильнее сказать «почти любой»). Но это слишком сложно, чтобы изучать в курсе школьной информатики). Такая цветовая модель называется RGB. В ней для указания цвета используются яркости (интенсивности свечения) красного (*Red*), зеленого (*Green*) и синего (*Blue*) цветов. Каждая яркость может принимать значение от нуля (отсутствие свечения) до 255 (максимальное свечение). То есть всего по 256 возможных состояний.

Вы, вероятно, понимаете, что количество цветов (равновероятных событий) и количество бит для хранения кода цвета (количество информации) связано соотношением

$$2^{\text{число бит}} = \text{число цветов} \quad (3.1)$$

То есть для хранения интенсивности каждого базового цвета требуется 8 бит ($2^{\text{число бит}} = 256$) или 1 байт. Значит, для хранения кода всего цвета требуется $3 \cdot 8 = 24$ бита или 3 байта. Заметим, что 2^{24} как раз примерно равно 16 миллионам.

В цветовой модели RGB работают все современные мониторы и телевизоры. В этой же цветовой модели представляется цвет в различных программах. В частности, для указания цвета фона web-страницы (в файле формата HTML) тоже используется цветовая модель RGB. Для этого используется атрибут bgcolor тега <body>, который записывается в виде #RRGGBB. Здесь RR — шестнадцатеричный код красной цветовой компоненты, GG — шестнадцатеричный код зеленой цветовой компоненты, BB — шестнадцатеричный код синей цветовой компоненты. Напоминаем, чем больше значение компоненты, тем больше интенсивность свечения соответствующего базового цвета. 00 — отсутствие свечения, FF — максимальное свечение ($FF_{16} = 255_{10}$). Рассмотрим коды основных цветов, получающиеся из этих соображений:

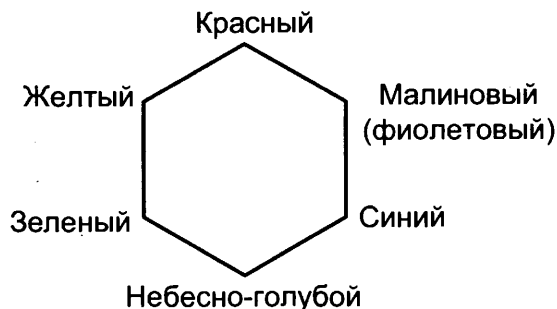
- #FF0000 красный (красная составляющая максимальная, остальные равны нулю)
- #000000 черный (ни одна компонента не светится)
- #FFFFFF белый (все составляющие максимальны и одинаковы, наиболее яркий цвет)
- #404040 темно-серый (все составляющие одинаковы, значит, ни один из базовых цветов не выделяется из остальных (серый); число 40_{16} меньше, чем среднее значение яркости (80_{16}), значит, это темный цвет.

3.16. Установите соответствие между кодами цвета и их названиями (соедините линиями).

Код цвета	Название цвета
а) #0000FF	1) средней яркости серый
б) #00FF00	2) почти черный
в) #808080	3) темно-синий
г) #C0C0C0	4) светло-серый
д) #010101	5) синий
е) #000080	6) зеленый

Напоминание. Принципы смешения базовых цветов отличаются от тех принципов смешения красок, которые привиты каждому ребенку на уроках рисования. Если смешать красный и синий, то все же получится фиолетовый. А вот если смешать желтый и синий, то получится вовсе не зеленый, а

белый цвет. (Различие связано с тем, что на рисовании смешиваются краски, а не самосветящиеся источники. Для красок физический принцип получения цвета и другая цветовая модель — CMYK.) Чтобы попытаться понять эти принципы, рекомендуем запомнить схему смешения цветов:



Если смешиваются две базовые компоненты с равными яркостями, получится цвет, расположенный на схеме между ними, светлость/темнота которого зависит от этих яркостей (чем ярче, тем светлее, чем меньше яркость, тем темнее).

Если нужно получить, например, светло-красный — очевидно, что нужно увеличить общую яркость по сравнению с красным цветом. Увеличивать яркость красной компоненты (#FF0000) уже некуда (она максимальна). Если увеличить одну из оставшихся компонент — в цвете будет присутствовать этот оттенок. Значит, нужно увеличить яркости обеих оставшихся компонент — #FF8080.

3.17. Установите соответствие между кодами цвета и их названиями (соедините линиями)

Код цвета	Название цвета
а) #FFFF00	1) темно-малиновый
б) #FFF800	2) малиновый
в) #40FF40	3) желтый
г) #FF00FF	4) светло-желтый
д) #800080	5) светло-зеленый

Если две компоненты смешиваются в разных пропорциях, получится оттенок между базовым цветом и тем, который указан между ними. Самый понятный пример — оранжевый цвет. Все понимают, что это нечто среднее между желтым и красным.

Желтый — #FFFF00, красный — #FF0000. Можно просто найти среднее арифметическое по каждой компоненте в отдельности: $R = (FF + FF) / 2 = FF$, $G = (FF + 00) / 2 \approx 80$, $B = (0 + 0) / 2 = 0$. То есть оранжевый — #FF8000.

3.18. Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#FFFFFF">`?

- 1) Белый 2) Зеленый 3) Красный 4) Синий

3.19. Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#404000">`?

- 1) Серый 2) Фиолетовый 3) Темно-красный 4) Коричневый

3.20. Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor = "#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor = "#800080">`?

- 1) Серый 2) Фиолетовый 3) Темно-красный 4) Коричневый

Напоминание. Растровое изображение состоит из прямоугольного набора точек (пикселей), для каждой из которых хранится код цвета этой точки.

Формула соотношения количества бит, выделенных для хранения кода цвета и количества цветов, которые могут быть таким образом закодированы, мы уже приводили:

$$2^{\text{число бит}} = \text{число цветов}$$

3.21. Какой минимальный объем памяти (в битах) необходим для хранения одной точки изображения, в котором 16 различных цветов?

Ответ:																			

3.22. Какой минимальный объем памяти (в битах) необходим для хранения одной точки изображения, в котором 2 различных цвета?

Ответ:																			

3.23. Какой минимальный объем памяти (в битах) необходим для хранения одной точки изображения, в котором 256 различных цветов?

Ответ:																			

3.24. Какой минимальный объем памяти (в битах) необходим для хранения одной точки изображения, в котором 32 различных цвета?

Ответ:																			

3.25. Какой минимальный объем памяти (в битах) необходим для хранения одной точки изображения, в котором 128 различных цветов?

Ответ:																			

3.26. Какое наибольшее количество различных цветов можно закодировать, используя 4 бита?

Ответ:																			

Рекомендация. Задача может быть решена двумя способами: можно увидеть, что общее количество 5-буквенных слов, составленных из трех букв, составляет $3^5 = 243$ и выписывать слова от конца списка. Но это сработает только в том случае, если требуемое по порядку слово близко к концу последовательности (как в этом примере). Существует более универсальный способ. Его идея состоит в том, что такая последовательность слов, записанных в алфавитном порядке при помощи ограниченного количества символов, аналогична системе счисления. Так как в данном случае используется 3 буквы, это аналогично троичной системе счисления. Поставим в соответствие буквам цифры. Самой первой в алфавитном порядке букве сопоставим 0, следующей — 1, следующей — 2. Получим такую последовательность:

1. ААААА — 00000
2. ААААО — 00001
3. ААААУ — 00002
4. АААОА — 00010

При этом заметим, что на позиции 1 стоит число 00000, то есть 0, а на позиции 2 — число 00001, то есть 1. Значит, на позиции 240 стоит число 239, записанное в троичной системе счисления. Переведем число 239 в троичную систему (например, будем делить уголком и собирать остатки). Получится 22212_3 . Остается только записать вместо каждой цифры соответствующую букву.

3.46. Все 4-буквенные слова, составленные из букв К, Л, М, Н, записаны в алфавитном порядке.

Вот начало списка:

1. КККК
2. КККЛ
3. КККМ
4. КККН
5. ККЛК

.....

Запишите слово, которое стоит на 215-м месте от начала списка.

3.47. Все 4-буквенные слова, составленные из букв П, Р, С, Т, записаны в алфавитном порядке.

Вот начало списка:

1. ПППП
2. ПППР
3. ПППС
4. ПППТ
5. ППРП

.....

Запишите слово, которое стоит на 205-м месте от начала списка.

3.48. По каналу связи передаются сообщения, каждое из которых содержит 16 букв А, 8 букв Б, 4 буквы В и 4 буквы Г (других букв в сообщениях нет). Каждую букву кодируют двоичной последовательностью. При выборе кода учитывались два требования:

- а) ни одно кодовое слово не является началом другого (это нужно, чтобы код допускал однозначное декодирование);
- б) общая длина закодированного сообщения должна быть как можно меньше.

Какой код из приведенных ниже следует выбрать для кодирования букв А, Б, В и Г?

- 1) А:0, Б:10, В:110, Г:111
- 2) А:0, Б:10, В:01, Г:11
- 3) А:1, Б:01, В:011, Г:001
- 4) А:00, Б:01, В:10, Г:11

Решение. Способ 1 (в лоб). Для каждого варианта проверить выполнение условия 1 и посчитать длину из условия 2. Условие 1 проверяем так: берем все кодовые слова не максимальной длины и смотрим, не стоят ли они в начале кодов большей длины. В данном примере этому условию не удовлетворяет вариант 2 (кодированное слово «0»/А является началом кодированного слова «01»/В), а также вариант 3 (кодированное слово «01»/Б является началом кодированного слова «011»/В). Для оставшихся вариантов вычисляем суммарную длину сообщения. Вариант 1: $16 \cdot 1 + 8 \cdot 2 + 4 \cdot 3 + 4 \cdot 3 = 16 + 16 + 12 + 12 = 56$ (16 букв А длины 1 + 8 букв В длины 2 + 4 буквы В длины 3 + 4 буквы Г длины 3). Вариант 2: $16 \cdot 2 + 8 \cdot 2 + 4 \cdot 2 + 4 \cdot 2 = 32 + 16 + 8 + 8 = 64$. Выбираем вариант 1 как более короткий.

Способ 2. Вычислим количество информации в каждом символе. Всего букв в сообщении $16 + 8 + 4 + 4 = 32$. Вычисляем вероятности каждой буквы. А: $16/32 = 1/2$, Б: $8/32 = 1/4$, В: $4/32 = 1/8$, Г: $4/32 = 1/8$. Применяем формулу вероятностного подхода к измерению информации $i = -\log_2 p$ (буквы-события не равновероятны, поэтому использовать формулу $2^i = N$ нельзя). Получаем: А: 1, Б: 2, В: 3, Г: 3.

Для того, чтобы длина сообщения была наименьшей, длины кодовых слов должны быть равны количеству информации в соответствующем символе. (Вообще говоря, это верно только при таком случае, как здесь — когда вероятности появления букв являются степенями двойки).

Такие длины кодовых слов есть только в варианте 1 и варианте 3. Проверяем оба варианта на выполнение условия 1. Вариант 3 не подходит (кодированное слово «01»/Б является началом кодированного слова «011»/В). Получаем, что верный *ответ* — вариант 1.

3.49. По каналу связи передаются сообщения, каждое из которых содержит 64 буквы А, 32 буквы Б, 16 букв В и 16 букв Г (других букв в сообщениях нет). Каждую букву кодируют двоичной последовательностью. При выборе кода учитывались два требования:

- а) ни одно кодовое слово не является началом другого (это нужно, чтобы код допускал однозначное декодирование);
- б) общая длина закодированного сообщения должна быть как можно меньше.

Какой код из приведенных ниже следует выбрать для кодирования букв А, Б, В и Г?

- 1) А:11, Б:10, В:01, Г:00
- 2) А:0, Б:10, В:100, Г:110
- 3) А:1, Б:01, В:001, Г:000
- 4) А:1, Б:01, В:10, Г:00

3.50. По каналу связи передаются сообщения, каждое из которых содержит 8 букв А, 8 букв Б, 16 букв В и 32 буквы Г (других букв в сообщениях нет). Каждую букву кодируют двоичной последовательностью. При выборе кода учитывались два требования:

- а) ни одно кодовое слово не является началом другого (это нужно, чтобы код допускал однозначное декодирование);
- б) общая длина закодированного сообщения должна быть как можно меньше.

Какова минимально возможная длина закодированного сообщения?

3.51. По каналу связи передаются сообщения, каждое из которых содержит 100 букв А, 50 букв Б, 25 букв В и 25 букв Г (других букв в сообщениях нет). Каждую букву кодируют двоичной последовательностью. При выборе кода учитывались два требования:

- а) ни одно кодовое слово не является началом другого (это нужно, чтобы код допускал однозначное декодирование);
- б) общая длина закодированного сообщения должна быть как можно меньше.

Какова минимально возможная длина закодированного сообщения?

3.52. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 30 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б; пропускная способность канала связи с городом Б в 4 раза выше, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б? В ответе запишите только целое число, единицу измерения писать не нужно.

Решение. Формула для вычисления объема звукового файла: $V = Ch \cdot v \cdot (i / 8) \cdot t$ (Число каналов, умноженное на частоту дискретизации, на время записи, на разрешение). Так как разрешение стало в 2 раза выше, а частота дискретизации в 1,5 раза меньше, то объем передаваемых данных стал в 2 раза больше и при этом в 1,5 раза меньше. То есть, изменился в $2/1,5 = 4/3$ раза.

Так как пропускная способность канала связи стала в 4 раза выше, то скорость передачи информации стала в 4 раза больше. А значит, время передачи информации будет в 4 раза меньше.

Получаем: исходное время передачи информации (30 секунд) изменится в $\frac{4}{3 \cdot 4}$ раза. Получаем:

$$\text{ем: } 30 \cdot \frac{4}{3 \cdot 4} = 10 \text{ секунд.}$$

3.53. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 40 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 1,5 раза выше и частотой дискретизации в 2 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б; пропускная способность канала связи с городом Б в 2 раза выше, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б? В ответе запишите только целое число, единицу измерения писать не нужно.

3.54. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 50 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 3 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б; пропускная способность канала связи с городом Б в 2 раза выше, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б? В ответе запишите только целое число, единицу измерения писать не нужно.

3.55. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 30 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 1,5 раза ниже и частотой дискретизации в 2 раза больше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б; пропускная способность канала связи с городом Б в 2 раза ниже, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б? В ответе запишите только целое число, единицу измерения писать не нужно.

4. ОСНОВЫ ЛОГИКИ

Основные понятия

В алгебре логики изучаются логические операции, производимые над высказываниями. Высказывания могут быть истинными или ложными. Применяя к простым высказываниям логические операции, можно строить составные высказывания.

Основными логическими операциями являются:

1. Отрицание (инверсия, логическое НЕ)

Смысл операции: результат меняется на противоположный (вместо истины — ложь, вместо лжи — истина).

Обозначение¹: \neg

Таблица истинности:

A	$\neg A$
0	1
1	0

2. Логическое сложение (дизъюнкция, логическое ИЛИ)

Смысл операции: результат — истина, если хотя бы один операнд — истина (операндом называется то значение или та переменная, над которым (которой) осуществляется операция).

Обозначение: \vee

Другое обозначение: $+$

Таблица истинности:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

3. Логическое умножение (конъюнкция, логическое И)

Смысл операции: результат — истина, если оба операнда — истина.

Обозначение: \wedge

Другие обозначения: $\&$, \cdot

Таблица истинности:

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Эти три операции считаются базовыми. Остальные рассмотренные нами операции являются дополнительными. Их всегда можно выразить через базовые операции.

¹ В различных учебниках информатики обозначения и терминология иногда отличаются. Здесь и далее мы будем применять обозначения и терминологию, используемую в КИМ ЕГЭ.

4. **Следование** (импликация)

Смысл операции: из лжи может следовать что угодно, а из истины — только истина.

Обозначение: \rightarrow

Таблица истинности:

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

5. **Равносильность** (эквиваленция)

Смысл операции: результат — истина, если операнды одинаковые.

Обозначение: \equiv

Другое обозначение: \leftrightarrow

Таблица истинности:

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

6. **Сложение по модулю 2** (исключающее ИЛИ, в просторечье XOR)

Смысл операции: результат — истина, если операнды разные.

Обозначение: \oplus

Таблица истинности:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Если в логическом выражении используется несколько логических операций, их порядок определяется **приоритетами логических операций**:

0) выражение в скобках,

1) логическое НЕ (инверсия),

2) логическое И (конъюнкция),

3) логическое ИЛИ (дизъюнкция), исключающее ИЛИ (эти две операции имеют одинаковый приоритет и выполняются слева направо),

4) следование (импликация),

5) равносильность (эквиваленция).

Как вы, вероятно, поняли, для изменения порядка выполнения логических операций используются скобки.

Выражение дополнительных логических операций через базовые операции

Импликация:

$$A \rightarrow B = \neg A \vee B$$

Эквиваленция:

$$A \equiv B = (A \wedge B) \vee (\neg A \wedge \neg B)$$

Исключающее ИЛИ:

$$A \oplus B = (\neg A \wedge B) \vee (A \wedge \neg B)$$

Основные законы алгебры логики

Название закона	Формулировка
Переместительный закон	$A \vee B = B \vee A$ $A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C)$ $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
Закон непротиворечия. Этот закон выражает тот факт, что высказывание не может быть одновременно истинным и ложным	$A \wedge \neg A = 0$
Закон исключенного третьего. Этот закон означает, что либо высказывание, либо его отрицание должно быть истинным	$A \vee \neg A = 1$
Закон двойного отрицания	$\neg(\neg A) = A$
Законы де Моргана	$\neg(A \vee B) = \neg A \wedge \neg B$ $\neg(A \wedge B) = \neg A \vee \neg B$
Законы переменной с самой собой	$A \vee A = A$ $A \wedge A = A$
Законы нуля и единицы	$A \wedge 0 = 0$ $A \wedge 1 = A$ $A \vee 0 = A$ $A \vee 1 = 1$
Законы поглощения	$A \vee (A \wedge B) = A$ $A \wedge (A \vee B) = A$ $A \vee (\neg A \wedge B) = A \vee B$

Практическая часть

4.1. Расставьте над символами логических операций их номера в порядке выполнения операций при вычислении выражения.

а) $\neg A \vee (B \vee \neg C)$

в) $A \wedge \neg (B \vee \neg C) \vee D$

б) $A \vee B \wedge C$

г) $\neg (A \vee B) \rightarrow C$

4.2. Заполните таблицу истинности логических выражений.

а)

A	B	A ∧ B	¬ (A ∧ B)

б)

A	B	C	A ∨ B	(A ∨ B) → C	¬ ((A ∨ B) → C)

в)

A	B	C	¬B	A ∨ ¬B	A ∨ ¬B ∨ C

г)

A	B	C	¬A	¬B	¬A ∧ ¬B	¬A ∧ ¬B ∧ C

б) $\neg ((L \rightarrow M) \rightarrow (L \wedge N))$

Ответ:																											

в) $(L \vee M) \rightarrow (\neg L \wedge N)$

Ответ:																											

г) $(L \vee \neg M) \rightarrow N$

Ответ:																											

Пример. Найдите все целые числа X , для которых истинно высказывание:

$$(X > 5) \wedge (X^2 \leq 49)$$

Решение:

Данное составное высказывание истинно, если одновременно истинны оба элементарных высказывания:

$$X > 5 \text{ и } X^2 \leq 49.$$

Решив эту элементарную систему неравенств (например, графически), получаем ответ.

Ответ: 6; 7.

4.6. Найдите все целые числа X , для которых истинно высказывание:

а) $(X > 5) \wedge (X + 3 < 11)$

Ответ:																											

б) $\neg (X > 1) \wedge (X^2 < 40)$

Ответ:																											

в) $\neg ((X > 5) \rightarrow (X > 8))$

Ответ:																											

г) $\neg ((X < 6) \rightarrow (X < 2))$

Ответ:																											

Задания в формате ЕГЭ

Пример 1. Какое логическое выражение равносильно выражению $\neg(A \vee B) \vee \neg C$?

- 1) $(\neg A \wedge \neg B) \vee \neg C$ 3) $A \vee B \wedge C$
2) $\neg A \vee \neg B \vee \neg C$ 4) $(A \wedge B) \vee C$

Решение.

Логические выражения называются равносильными, если при любых значениях входящих в них переменных значения этих выражений равны.

Преобразуем выражение $\neg(A \vee B)$ в соответствии с законом де Моргана: $\neg(A \vee B) = (\neg A \wedge \neg B)$, поэтому правилен ответ под номером 1.

Пример 2. Какое логическое выражение равносильно выражению $(A \vee \neg C) \wedge \neg(\neg B \wedge C)$?

- 1) $A \vee \neg C \wedge B \vee \neg C$ 3) $(A \wedge B) \vee \neg C$
2) $A \vee \neg C$ 4) $\neg C \wedge (A \vee B)$

Решение:

Преобразуем выражение $\neg(\neg B \wedge C)$ в соответствии с законом де Моргана: $\neg(\neg B \wedge C) = \neg(\neg B) \vee \neg C$. Избавимся от двойного отрицания: $= (B \vee \neg C)$. Получаем общее выражение: $(A \vee \neg C) \wedge (B \vee \neg C)$. Но такого выражения нет среди вариантов ответа. Заметим, что в обеих скобках используются одинаковые операции (дизъюнкция) и имеется одинаковая переменная ($\neg C$). Вынесем эту переменную за скобки (по распределительному закону): $(A \wedge B) \vee \neg C$. Это выражение стоит третьим среди вариантов ответа.

4.7.

а) Какое логическое выражение равносильно выражению $\neg(\neg A \vee B) \vee \neg C$?

- 1) $(A \wedge \neg B) \vee \neg C$ 2) $\neg A \vee B \vee \neg C$ 3) $A \vee \neg B \wedge C$ 4) $(\neg A \wedge B) \vee C$

б) Какое логическое выражение равносильно выражению $(\neg A \vee B) \vee C$?

- 1) $(A \wedge \neg B) \vee C$ 2) $\neg A \vee B \vee C$ 3) $A \vee \neg B \vee C$ 4) $(\neg A \wedge B) \vee C$

в) Какое логическое выражение равносильно выражению $\neg(A \vee B \vee C)$?

- 1) $A \vee B \vee C$ 2) $\neg A \vee \neg B \vee \neg C$ 3) $A \wedge B \wedge C$ 4) $\neg A \wedge \neg B \wedge \neg C$

г) Какое логическое выражение равносильно выражению $(\neg A \vee \neg B) \vee \neg C$?

- 1) $\neg(A \wedge B) \vee \neg C$ 2) $\neg A \wedge \neg B \vee \neg C$ 3) $\neg A \wedge (\neg B \wedge \neg C)$ 4) $(A \wedge B) \wedge C$

4.8.

а) Какое логическое выражение равносильно выражению $\neg(A \wedge \neg C) \wedge B$?

- 1) $(\neg A \vee C) \wedge B$ 2) $\neg A \wedge B \vee \neg C$ 3) $(A \wedge \neg C) \wedge \neg B$ 4) $\neg A \vee B \wedge C$

б) Какое логическое выражение равносильно выражению $\neg A \wedge \neg(\neg B \wedge C)$?

- 1) $\neg A \wedge B \wedge \neg C$ 2) $\neg A \wedge \neg C \vee B$ 3) $\neg A \wedge (B \vee \neg C)$ 4) $(\neg A \wedge B) \wedge C$

в) Какое логическое выражение равносильно выражению $\neg(A \wedge \neg B) \wedge \neg C$?

- | | |
|----------------------------------------------------|-------------------------------|
| 1) $(\neg A \vee B) \vee \neg C$ | 3) $(\neg A \vee B) \vee C$ |
| 2) $(B \wedge \neg C) \vee (\neg A \wedge \neg C)$ | 4) $(\neg A \vee B) \wedge C$ |

г) Какое логическое выражение равносильно выражению $C \wedge \neg(\neg A \wedge B)$?

- | | |
|----------------------------------|------------------------------------------|
| 1) $(A \vee \neg B) \vee \neg C$ | 3) $(\neg A \vee B) \vee C$ |
| 2) $(\neg A \vee B) \wedge C$ | 4) $(\neg B \wedge C) \vee (A \wedge C)$ |

4.9.

а) Какое логическое выражение равносильно выражению $\neg(A \wedge \neg C) \wedge B$?

- | | |
|----------------------------------|--------------------------------------|
| 1) $\neg A \vee C \wedge B$ | 3) $(A \wedge \neg C) \wedge \neg B$ |
| 2) $\neg A \wedge B \vee \neg C$ | 4) $\neg A \wedge B \vee C \wedge B$ |

б) Какое логическое выражение равносильно выражению $(\neg A \vee \neg C) \wedge \neg(\neg B \wedge C)$?

- | | |
|----------------------------------------------|------------------------------------|
| 1) $\neg A \vee B \wedge \neg C$ | 3) $(\neg A \vee B) \wedge \neg C$ |
| 2) $\neg A \vee \neg C \wedge B \vee \neg C$ | 4) $\neg A \wedge B \vee \neg C$ |

в) Какое логическое выражение равносильно выражению $\neg(A \wedge \neg B) \wedge \neg(A \wedge C)$?

- | | |
|----------------------------------------------|--------------------------------------------------|
| 1) $\neg(A \wedge \neg B \wedge A \wedge C)$ | 3) $\neg(\neg A \vee B \vee \neg A \vee \neg C)$ |
| 2) $\neg A \vee B \wedge \neg C$ | 4) $\neg A \vee B \wedge \neg A \vee \neg C$ |

г) Какое логическое выражение равносильно выражению $(A \wedge C) \vee (A \wedge \neg C) \vee \neg(\neg A \rightarrow \neg B)$?

- | | |
|--------------------------|---------------|
| 1) 1 | 3) $A \vee B$ |
| 2) $A \wedge (C \vee B)$ | 4) B |

4.10.

а) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	0
1	1	0	1
1	0	0	1

Какое выражение соответствует F?

- | | | | |
|-------------------------------------|------------------------------------|----------------------|--------------------------|
| 1) $\neg X \vee \neg Y \vee \neg Z$ | 2) $X \wedge \neg Y \wedge \neg Z$ | 3) $X \vee Y \vee Z$ | 4) $X \wedge Y \wedge Z$ |
|-------------------------------------|------------------------------------|----------------------|--------------------------|

б) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	1
1	1	0	1
1	0	0	1

Какое выражение соответствует F?

- | | | | |
|-----------------------------------------|------------------------------------|---------------------------|---------------------------|
| 1) $\neg X \wedge \neg Y \wedge \neg Z$ | 2) $X \wedge \neg Y \wedge \neg Z$ | 3) $X \vee Y \vee \neg Z$ | 4) $\neg X \vee Y \vee Z$ |
|-----------------------------------------|------------------------------------|---------------------------|---------------------------|

- в) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	0
0	1	0	1
1	0	0	0

Какое выражение соответствует F?

- 1) $\neg X \wedge Y \wedge \neg Z$ 2) $X \wedge \neg Y \wedge \neg Z$ 3) $X \vee Y \vee \neg Z$ 4) $\neg X \vee Y \vee Z$

- г) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	1	0	0
1	1	1	0
1	0	0	1

Какое выражение соответствует F?

- 1) $\neg X \wedge Y \wedge \neg Z$ 2) $X \wedge \neg Y \wedge \neg Z$ 3) $X \vee Y \vee \neg Z$ 4) $\neg X \vee Y \vee Z$

4.11.

- а) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	1	1
0	1	0	0
1	0	1	1

Какое выражение соответствует F?

- 1) $\neg X \vee \neg Y \wedge Z$ 2) $\neg X \vee Y \vee Z$ 3) $\neg X \wedge \neg Y \vee \neg Z$ 4) $X \wedge \neg Y \vee Z$

- б) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	1	0	1
1	1	0	1
0	0	1	0

Какое выражение соответствует F?

- 1) $\neg X \vee Y \wedge \neg Z$ 2) $\neg X \vee Y \vee Z$ 3) $X \wedge \neg Z \vee Y$ 4) $\neg X \wedge \neg Z \vee \neg Y$

- в) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: A, B, C. Дан фрагмент таблицы истинности выражения F:

A	B	C	F
0	1	1	1
1	0	1	1
0	1	0	0

Какое выражение соответствует F?

- 1) $\neg A \wedge B \wedge C$ 2) $A \vee B \wedge C$ 3) $\neg A \wedge B \vee \neg C$ 4) $\neg A \vee B \vee \neg C$

- г) Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: A, B, C. Дан фрагмент таблицы истинности выражения F:

A	B	C	F
0	1	1	1
1	1	0	0
0	1	0	1

Какое выражение соответствует F?

- 1) $\neg A \wedge B \wedge \neg C$ 2) $\neg A \vee B \vee \neg C$ 3) $A \vee B \wedge C$ 4) $\neg A \wedge B \vee C$

4.12.

- а) Дан фрагмент таблицы истинности выражения F:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
0	0	1	0	1	0	1	1
0	1	0	1	1	0	0	0
1	0	1	1	0	1	1	1

Каким из приведенных ниже выражений может быть F?

- 1) $\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6 \vee x_7$
 2) $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7$
 3) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee x_7$
 4) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge x_7$

- б) Дан фрагмент таблицы истинности выражения F:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	0
1	0	1	0	0	1	1	1

Каким из приведенных ниже выражений может быть F?

- 1) $\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6 \vee x_7$
 2) $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7$
 3) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee x_7$
 4) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge x_7$

в) Дан фрагмент таблицы истинности выражения F:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
0	0	1	0	1	0	1	1
0	1	0	1	1	0	0	1
1	0	1	1	0	1	1	0

Каким из приведенных ниже выражений может быть F?

- 1) $\neg x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6 \vee \neg x_7$
- 2) $\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7$
- 3) $\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee x_7$
- 4) $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge x_7$

г) Дан фрагмент таблицы истинности выражения F:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
0	0	1	0	1	0	1	0
0	1	0	1	1	0	0	0
1	0	1	1	0	1	1	0

Каким из приведенных ниже выражений может быть F?

- 1) $x_1 \vee \neg x_2 \vee x_3 \vee x_4 \vee \neg x_5 \vee x_6 \vee x_7$
- 2) $x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \wedge \neg x_5 \wedge x_6 \wedge x_7$
- 3) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee x_7$
- 4) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge x_7$

Пример. Найдите наименьшее целое число X, при котором истинно высказывание

$$((X > 7) \vee (X < 7)) \rightarrow (X > 8).$$

Решение.

Импликация истинна в трех случаях:

- 1) $0 \rightarrow 1$ (ложь \rightarrow истина)
- 2) $0 \rightarrow 0$ (ложь \rightarrow ложь)
- 3) $1 \rightarrow 1$ (истина \rightarrow истина)

Рассмотрим 1-й случай:

Предположим, что $(X > 7) \vee (X < 7) =$ ложь и $X > 8 =$ истина.

$(X > 7) \vee (X < 7) =$ ложь только при $X = 7$. $7 < 8$, поэтому в данном случае решения нет.

Рассмотрим 2-й случай:

Предположим, что $(X > 7) \vee (X < 7) =$ ложь и $X > 8 =$ ложь. В этом случае $X = 7$ является единственным решением.

Рассмотрим 3-й случай:

Предположим, что $(X > 7) \vee (X < 7) =$ истина, тогда X — любое целое число, кроме 7.

Из условия $X > 8$ получаем, что наименьшим решением будет 9. Но во втором случае мы получили решение 7. $7 < 9$, поэтому окончательный ответ: 7.

Второй вариант решения

Для начала приведем выражение в более удобную форму, вспомнив, что $(X > 7 \text{ ИЛИ } X < 7)$ в математике записывается как $X \neq 7$. $(X \neq 7) \rightarrow (X > 8)$.

Преобразуем данное выражение по законам алгебры логики.

Воспользуемся законом « $A \rightarrow B = \neg A \vee B$ ». $\neg(X \neq 7) \vee (X > 8) \Leftrightarrow (X = 7) \vee (X > 8)$. Таким образом, решением является число 7 и все числа больше 8. Наименьшим из этих чисел является число 7.

Заменяя выражения в первых скобках на эквиваленции, получим:

$$(x_1 \equiv x_2) \vee (x_1 \equiv x_3) = 1$$

$$(x_2 \equiv x_3) \vee (x_2 \equiv x_4) = 1$$

$$(x_3 \equiv x_4) \vee (x_3 \equiv x_5) = 1$$

$$(x_4 \equiv x_5) \vee (x_4 \equiv x_6) = 1$$

$$(x_5 \equiv x_6) \vee (x_5 \equiv x_7) = 1$$

Дальнейшее решение построим на принципе рассуждений. Будем предполагать, что переменная равна определенному значению, и смотреть, что из этого следует.

Пусть $x_1 = 1$.

Пусть $x_2 = 0$. Тогда первое условие первого уравнения не выполняется ($x_1 \equiv x_2$) и тогда должно выполняться второе условие первого уравнения ($x_1 \equiv x_3$). То есть $x_3 = 1$. Тогда получается, что $x_2 \neq x_3$ (не выполняется первое условие второго уравнения), значит, должно выполняться второе условие второго уравнения ($x_2 \equiv x_4$). Значит, $x_4 = 0$. Аналогично рассуждая, получаем, что значения последующих переменных должны чередоваться: $x_5 = 1$, $x_6 = 0$, $x_7 = 1$. Эта ветка рассуждений ($x_1 = 1$, $x_2 = 0$) привела нас к единственному решению. Запомним его (1).

Пусть $x_2 = 1$. Тогда первое условие первого уравнения выполняется ($x_1 \equiv x_2$). Это значит, что второе условие первого уравнения ($x_1 \equiv x_3$) не обязательно должно выполняться. То есть x_3 может быть любым.

Пусть $x_3 = 0$. Тогда первое условие второго уравнения ($x_2 \equiv x_3$) не выполняется. Значит, должно выполняться второе условие второго уравнения ($x_2 \equiv x_4$). То есть $x_4 = 1$. Ситуация аналогична уже рассмотренной нами для $x_2 = 0$ — значения последующих переменных должны чередоваться: $x_5 = 0$, $x_6 = 1$, $x_7 = 0$. Это единственное решение. Запомним его (1).

Пусть $x_3 = 1$. Тогда первое условие второго уравнения выполняется ($x_2 \equiv x_3$). Это значит, что второе условие второго уравнения ($x_2 \equiv x_4$) не обязательно должно выполняться. То есть x_4 может быть любым.

Мы пришли к повторяющейся ситуации. Для каждой последующей переменной, если она будет равна нулю, это будет давать единственное решение, а если единице, то нужно будет рассматривать два варианта значений следующей переменной. Получается, для каждой переменной $x_2 \dots x_7$ имеется единственное решение (с чередующимися значениями остальных переменных). Плюс у последней переменной (x_7) возможно два значения (2). Всего получается 7 возможных вариантов (для исключения ошибки рекомендуем аккуратно расписать все эти предположения для каждой переменной и потом посчитать «запомненные» (1) и (2)).

Нетрудно понять, что для $x_1 = 0$ ситуация совершенно аналогична (симметрична), и это дает нам еще 7 вариантов.

Ответ: 14.

4.17.

а) Сколько различных решений имеет система уравнений:

$$(x_1 \wedge x_2 \vee \neg x_1 \wedge \neg x_2) \vee (x_1 \equiv x_3) = 1$$

$$(x_2 \wedge x_3 \vee \neg x_2 \wedge \neg x_3) \vee (x_2 \equiv x_4) = 1$$

$$(x_3 \wedge x_4 \vee \neg x_3 \wedge \neg x_4) \vee (x_3 \equiv x_5) = 1$$

$$(x_4 \wedge x_5 \vee \neg x_4 \wedge \neg x_5) \vee (x_4 \equiv x_6) = 1$$

$$(x_5 \wedge x_6 \vee \neg x_5 \wedge \neg x_6) \vee (x_5 \equiv x_7) = 1$$

$$(x_6 \wedge x_7 \vee \neg x_6 \wedge \neg x_7) \vee (x_6 \equiv x_8) = 1$$

$$x_5 \equiv x_6 = 0$$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
			1		0			1
			0			0		1
0			1					0

Каким из приведенных ниже выражений может быть F?

- 1) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6 \wedge \neg x_7 \wedge \neg x_8$
- 2) $\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8$
- 3) $x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee x_5 \vee \neg x_6 \vee x_7 \vee x_8$
- 4) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7 \vee \neg x_8$

Рекомендация. Среди вариантов ответа обнаруживаем только либо сплошные конъюнкции, либо сплошные дизъюнкции. Смотрим на столбец результата (F) и видим там две единицы. Значит, это не может быть конъюнкцией. Значит, это дизъюнкция. Дизъюнкция дает в результате единственный ноль тогда и только тогда, когда складываются все нули. В строке 3 приведенной таблицы имеем ноль. Чтобы складывались все нули, нужно, чтобы переменные x_1 и x_4 (только для них указаны значения) были x_1 и $\neg x_4$ (переменная x_1 уже 0, а переменная x_4 равна 1, складывать нужно только нули, поэтому x_4 должна быть инвертирована). Ищем среди вариантов ответа дизъюнкцию (варианты 3 и 4), в которой x_1 и $\neg x_4$. Это вариант 4.

Если бы в столбце F были бы два нуля и одна единица, ответ нужно было бы искать среди конъюнкций, причем в той строке, к которой получается единственная 1, нужно чтобы все переменные были равны 1. Те переменные, которые в этой строке даны равными 0, нужно было бы инвертировать.

4.19.

- а) Миша заполнял таблицу истинности для выражения F. Он успел заполнить лишь небольшой фрагмент таблицы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
1			1					0
	0					1		1
		1			0			0

Каким из приведенных ниже выражений может быть F?

- 1) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6 \wedge \neg x_7 \wedge \neg x_8$
- 2) $\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8$
- 3) $\neg x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4 \vee x_5 \vee x_6 \vee x_7 \vee x_8$
- 4) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7 \vee \neg x_8$

- б) Миша заполнял таблицу истинности для выражения F. Он успел заполнить лишь небольшой фрагмент таблицы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
1			1					1
	0			1				0
		1			0			1

Каким из приведенных ниже выражений может быть F?

- 1) $\neg x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7 \vee x_8$
- 2) $x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7 \wedge \neg x_8$
- 3) $x_1 \vee x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7 \vee \neg x_8$
- 4) $\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8$

в) Миша заполнял таблицу истинности для выражения F. Он успел заполнить лишь небольшой фрагмент таблицы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
	1			1				0
		0			1			0
			1			0		1

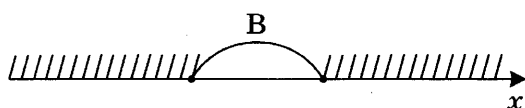
Каким из приведенных ниже выражений может быть F?

- 1) $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \wedge \neg x_5 \wedge \neg x_6 \wedge x_7 \wedge x_8$
- 2) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee \neg x_6 \vee \neg x_7 \vee \neg x_8$
- 3) $\neg x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7 \vee x_8$
- 4) $x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7 \wedge \neg x_8$

4.20. На числовой прямой дан отрезок $B = [4, 12]$. Выберите такой отрезок A, что формула $(x \in A) \vee \neg(x \in B)$ верна при любых значениях x .

- 1) [5, 10]
- 2) [0, 8]
- 3) [2, 15]
- 4) [20, 25]

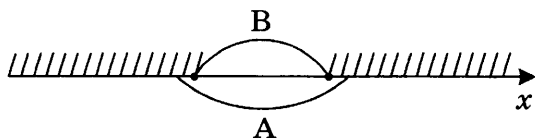
Решение. Нарисуем на числовой прямой область, которая нам известна. Это отрезок B. Найдем части числовой прямой, для которых приведенное условие точно выполняется. Так как логические выражения $(x \in A)$ и $\neg(x \in B)$ связаны дизъюнкцией, области, для которых результат будет истинным, должны объединяться. Значит, для всех точек числовой прямой, для которых одно из выражений будет истинным, общее выражение тоже будет истинным. Заштрихуем те точки числовой прямой, для которых выполняется условие $\neg(x \in B)$. Так как перед скобкой стоит отрицание, это будут все точки, которые НЕ входят в отрезок B:



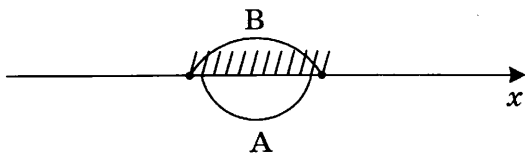
Незаштрихованными остались точки отрезка B. При объединении с отрезком A эти точки должны стать заштрихованными. Значит, все точки отрезка B должны целиком находиться внутри отрезка A. Осталось выбрать такой отрезок A, левая граница которого меньше левой границы отрезка B, а правая граница — больше правой границы отрезка B.

Разберем все возможные комбинации этой задачи:

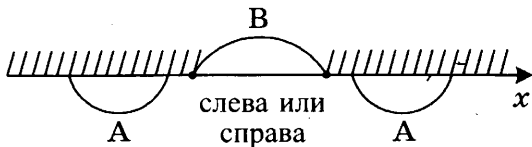
а) $(x \in A) \vee \neg(x \in B)$. Разобран выше. Отрезок B должен целиком находиться внутри отрезка A:



б) $\neg(x \in A) \vee (x \in B)$. Известно, что для всех точек отрезка B формула истинна. Значит, нужно, чтобы « $\neg A$ » «покрыло» все оставшиеся незаштрихованные области. Для этого отрезок A должен целиком находиться внутри отрезка B.



в) $\neg(x \in A) \vee \neg(x \in B)$. Заштрихованными оказываются все точки, не входящие в B. Значит, « $\neg A$ » должен «покрывать» отрезок B. Для этого нужно, чтобы ни одна точка отрезка A не находилась на отрезке B.



Если исходная формула не такая простая, нужно привести ее к этому виду. Для этого:

- 1) если встречается операция \rightarrow , преобразуйте ее по закону: $A \rightarrow B = \neg A \vee B$;
- 2) если отрицание стоит перед скобкой, в которой \wedge или \vee : примените закон де Моргана и уберите отрицание перед скобкой ($\neg(A \vee B) = \neg A \wedge \neg B$, $\neg(A \wedge B) = \neg A \vee \neg B$).

После этого:

- 1) нарисуйте на числовой прямой оставшиеся области,
- 2) заштрихуйте соответствующие им области (внутри отрезка, если нет отрицания, или снаружи отрезка, если есть отрицание),
- 3) области, объединенные операцией \vee , штрихуйте в одну сторону и считайте результатом все, куда попала хотя бы одна штриховка,
- 4) области, пересеченные операцией \wedge , штрихуйте в разную сторону и считайте результатом только то, где обе штриховки пересеклись.

В результате задача сведется к одному из случаев а), б) или в), разобранных нами ранее.

4.21. На числовой прямой дан отрезок $B = [4, 12]$. Выберите такой отрезок A, что формула $\neg(x \in A) \vee (x \in B)$ верна при любых значениях x .

- 1) [5, 10] 2) [0, 8] 3) [2, 15] 4) [20, 25]

4.22. На числовой прямой даны отрезки $B = [2, 12]$ и $C = [7, 18]$. Выберите такой отрезок A, что формула $(x \in A) \vee ((x \in B) \rightarrow (x \in C))$ верна при любых значениях x .

- 1) [3, 6] 2) [6, 14] 3) [1, 9] 4) [5, 20]

4.23. На числовой прямой даны отрезки $B = [2, 12]$ и $C = [7, 18]$. Выберите такой отрезок A, что формула $\neg(x \in A) \vee \neg((x \in B) \rightarrow \neg(x \in C))$ верна при любых значениях x .

- 1) [1, 20] 2) [6, 14] 3) [1, 9] 4) [8, 10]

4.24. На числовой прямой даны отрезки $B = [2, 12]$ и $C = [7, 18]$. Выберите такой отрезок A, что формула $(x \in A) \rightarrow \neg(\neg(x \in B) \wedge \neg(x \in C))$ верна при любых значениях x .

- 1) [1, 20] 2) [13, 16] 3) [1, 9] 4) [8, 20]

4.25. На числовой прямой даны отрезки $B = [2, 12]$ и $C = [7, 18]$. Выберите такой отрезок A, что формула $(\neg(x \in B) \rightarrow (x \in C)) \rightarrow \neg(x \in A)$ верна при любых значениях x .

- 1) [20, 25] 2) [8, 11] 3) [1, 20] 4) [6, 15]

- 4.26.** Каждое из логических выражений A и B зависит от одного и того же набора из 5 переменных. В таблицах истинности обоих выражений в столбцах значений стоит ровно по 4 единицы в каждой таблице. Каково минимально возможное число единиц в столбце значений таблицы истинности выражения $A \vee \neg B$?

Решение.

Таблица истинности содержит $2^5 = 32$ строки. Поскольку в таблице истинности значения могут быть только 0 или 1, в столбцах значений для выражения $\neg B$ содержится 28 единиц и 4 нуля. Выражение $A \vee \neg B$ истинно, если хотя бы одно из выражений A или $\neg B$ истинно. Минимально возможное число единиц в столбце значений таблицы истинности выражения $A \vee \neg B$ достигается, когда нулевые значения выражения B сочетаются с нулевыми значениями выражения A . Поскольку нулевых значений выражения A заведомо больше четырех, искомый минимум равен 28.

- 4.27.** Каждое из логических выражений A и B зависит от одного и того же набора из 8 переменных. В таблицах истинности обоих выражений в столбцах значений стоит ровно по 5 единиц в каждой таблице. Каково минимально возможное число нулей в столбце значений таблицы истинности выражения $A \wedge B$?

- 4.28.** Каждое из логических выражений A и B зависит от одного и того же набора из 7 переменных. В таблицах истинности обоих выражений в столбцах значений стоит ровно по 10 единиц в каждой таблице. Каково минимально возможное число нулей в столбце значений таблицы истинности выражения $A \wedge B$?

- 4.29.** Каждое из логических выражений A и B зависит от одного и того же набора из 7 переменных. В таблицах истинности обоих выражений нет ни одной совпадающей строки. Сколько единиц будет содержаться в столбце значений таблицы истинности выражения $A \wedge B$?

- 4.30.** Каждое из логических выражений A и B зависит от одного и того же набора из 7 переменных. В таблицах истинности обоих выражений нет ни одной совпадающей строки. Сколько единиц будет содержаться в столбце значений таблицы истинности выражения $A \vee B$?

- 4.31.** Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наибольшего натурального числа A формула

$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 4))$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Решение.

Упростим выражение. Избавимся от операции импликация, смысл которой с точки зрения арифметики выразить затруднительно. По формуле $A \rightarrow B = \neg A \vee B$ получаем:

$$\text{ДЕЛ}(x, A) \vee (\neg \text{ДЕЛ}(x, 6) \vee \neg \text{ДЕЛ}(x, 4))$$

Так как смысл выражения $(\neg \text{ДЕЛ}(x, 6) \vee \neg \text{ДЕЛ}(x, 4))$ все еще трудно понять, применим закон Де Моргана (вынесем отрицание за скобки). Получим $\neg(\text{ДЕЛ}(x, 6) \wedge \text{ДЕЛ}(x, 4))$. Выражение в скобках означает, что число x одновременно делится на 6 и на 4. Это значит, что оно делится на наименьшее общее кратное чисел 6 и 4. То есть, на число 12. Получаем общее выражение:

$$\text{ДЕЛ}(x, A) \vee \neg \text{ДЕЛ}(x, 12)$$

Выражение $\neg \text{ДЕЛ}(x, 12)$ дает нам все числа, которые не делятся на 12. Так как общее выражение должно быть верно для всех натуральных x , к этим числам не хватает только тех, которые делятся на 12. Выражение $\text{ДЕЛ}(x, A)$ — все числа, делящиеся на A . Значит, нужное нам наибольшее значение A равно 12.

4.32. Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $11 \& 6 = 1011_2 \& 0110_2 = 0010_2 = 2$.

Для какого наибольшего неотрицательного целого числа A формула

$$x \& A \neq 0 \rightarrow (x \& 10 = 0 \rightarrow x \& 6 \neq 0)$$

тождественно истинна (т.е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Решение.

Упростим выражение. Избавимся от операции импликация. Получаем:

$$x \& A = 0 \vee (x \& 10 \neq 0 \vee x \& 6 \neq 0)$$

Проанализируем, что означает выражение $x \& 10 \neq 0$. $10 = 1010_2$. После поразрядной конъюнкции числа x с числом 1010_2 из двоичного представления числа x все его двоичные цифры станут равны 0, кроме тех, которые стоят на позициях единиц в числе 1010_2 . То есть, результат будет не равен нулю, если в двоичном представлении числа x на позициях номер 1 и 3 (нумерация справа налево начиная с нуля) стоит хотя бы одна единица. А для выражения $x \& 6 \neq 0$ это будет для позиций 1 и 2 ($6 = 110_2$).

Значит, выражение $(x \& 10 \neq 0 \vee x \& 6 \neq 0)$ будет истинно, если в двоичном представлении числа x есть хотя бы одна единица на позициях 1, 2 или 3. Это можно иначе записать как $x \& 1110_2 \neq 0$. То есть, $x \& 14 \neq 0$. Получаем выражение: $x \& A = 0 \vee x \& 14 \neq 0$. Правая часть выражения истинна, если хотя бы на одной из позиций 1, 2 или 3 в числе x стоит единица. Значит, чтобы выражение было истинно для любых x , нужно добавить все такие числа, у которых на этих позициях стоят нули. Наибольшее число, которое может быть подставлено в условие $x \& A = 0$ для достижения такого результата — это само число 14.

4.33. Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наибольшего натурального числа A формула

$$\neg (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9)) \rightarrow \text{ДЕЛ}(x, A)$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

4.34. Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $11 \& 6 = 1011_2 \& 0110_2 = 0010_2 = 2$.

Для какого наибольшего неотрицательного целого числа A формула

$$x \& A \neq 0 \rightarrow (x \& 12 = 0 \rightarrow x \& 5 \neq 0)$$

тождественно истинна (т.е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

4.35. Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $11 \& 6 = 1011_2 \& 0110_2 = 0010_2 = 2$.

Для какого наибольшего неотрицательного целого числа A формула

$$\neg (x \& 3 = 0 \rightarrow x \& 9 \neq 0) \rightarrow x \& A = 0$$

тождественно истинна (т.е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

В ответе запишите число — количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).

Например, нижний левый угол лабиринта имеет адрес А1.

Решение.

Сначала определим свойства клеток, отвечающих требованию задачи.

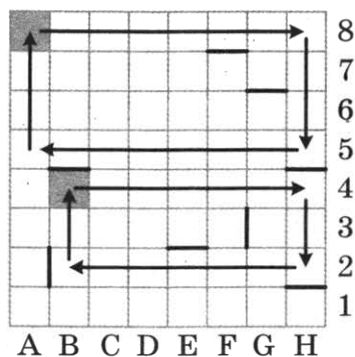
Поскольку последний цикл программы РОБОТА

ПОКА сверху свободно

ДЕЛАТЬ вверх

искомые клетки должны иметь границу сверху.

Таких клеток 14: вся верхняя горизонталь и клетки F7, G6, B4, H4, E2 и H1.



Предпоследний цикл программы РОБОТА

ПОКА слева свободно

ДЕЛАТЬ влево

Это значит, что в вертикали искомой клетки ниже нее обязательно должна быть клетка с левой границей, от которой «отскочил» РОБОТ перед тем, как пойти вверх.

Из ранее найденных клеток этому условию удовлетворяют 4 — A8, B8, B4, G6.

При этом B8 надо тоже отбросить, поскольку в ее вертикали на пути от клетки B2 стоит горизонтальная граница между B4 и B5.

Далее используем остальные фрагменты программы. Выполнив программу целиком для трех оставшихся клеток, получаем, что условию задачи удовлетворяют только две из них — A8 и B4.

Ответ: 2, A8, B4.

5.3. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл

ПОКА < условие > команда

выполняется, пока условие истинно, иначе происходит переход на следующую строку. Если РОБОТ начнет движение в сторону стены, то он разрушится и программа прервется.

Нарисуйте путь РОБОТА в лабиринте, если он начал движение по программе из указанной клетки.

а)

Лабиринт	Программа	Начальная клетка
<p>6 5 4 3 2 1 A B C D E F</p>	<p>НАЧАЛО ПОКА < справа свободно > вправо ПОКА < снизу свободно > вниз ПОКА < слева свободно > влево ПОКА < сверху свободно > вверх КОНЕЦ</p>	В4

б)

Лабиринт	Программа	Начальная клетка
<p>6 5 4 3 2 1 A B C D E F</p>	<p>НАЧАЛО ПОКА < слева свободно > влево ПОКА < сверху свободно > вверх ПОКА < справа свободно > вправо ПОКА < снизу свободно > вниз КОНЕЦ</p>	Ф1

в)

Лабиринт	Программа	Начальная клетка
<p>6 5 4 3 2 1 A B C D E F</p>	<p>НАЧАЛО ПОКА < слева свободно > влево ПОКА < сверху свободно > вверх ПОКА < справа свободно > вправо ПОКА < снизу свободно > вниз КОНЕЦ</p>	Д3

г)

Лабиринт	Программа	Начальная клетка
<p>6 5 4 3 2 1 A B C D E F</p>	<p>НАЧАЛО ПОКА < слева свободно > влево ПОКА < справа свободно > вправо ПОКА < снизу свободно > вниз ПОКА < сверху свободно > вверх КОНЕЦ</p>	С5

5.4. Для исполнителя РОБОТ, описанного в предыдущем задании, составить программу не более чем из семи строк (включая строки «НАЧАЛО» и «КОНЕЦ»), обеспечивающую нужное перемещение по заданному лабиринту

а)

Лабиринт	Программа	Начальная клетка	Конечная клетка
		A1	F6

б)

Лабиринт	Программа	Начальная клетка	Конечная клетка
		C2	E5

в)

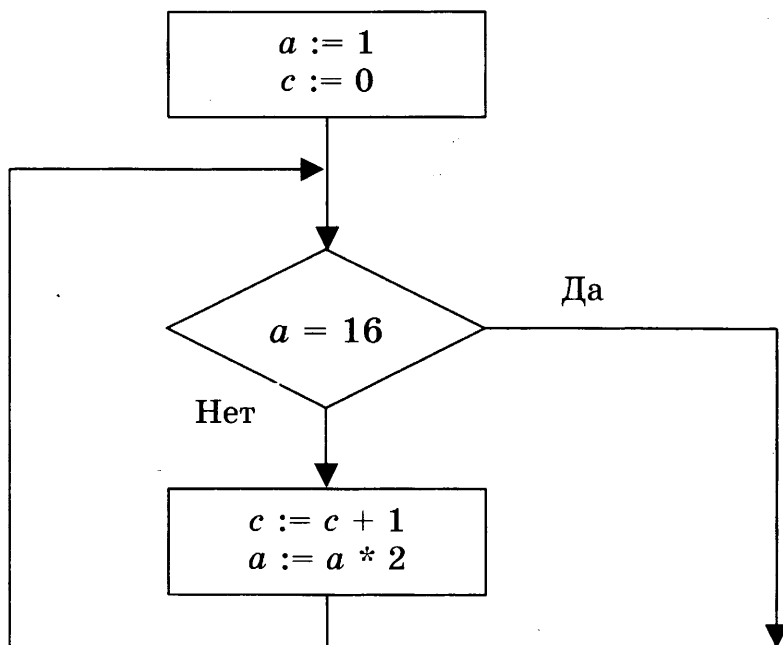
Лабиринт	Программа	Начальная клетка	Конечная клетка
		H1	D8

г)

Лабиринт	Программа	Начальная клетка	Конечная клетка
		D1	C2

Для решения задач на определение значения переменной после выполнения фрагмента программы или блок-схемы удобно использовать таблицу значений переменных.

Пример. Дана блок-схема алгоритма. Требуется найти значение переменной c после завершения алгоритма.



Решение.

1-й способ

Составим таблицу значений переменных, добавив в нее для удобства результаты вычисления логического выражения.

№ шага	Значение a	Значение c	$a = 16 ?$
0	1	0	нет
1	1	1	
2	2	1	нет
3	2	2	
4	4	2	нет
5	4	3	
6	8	3	нет
7	8	4	
7	16	4	да

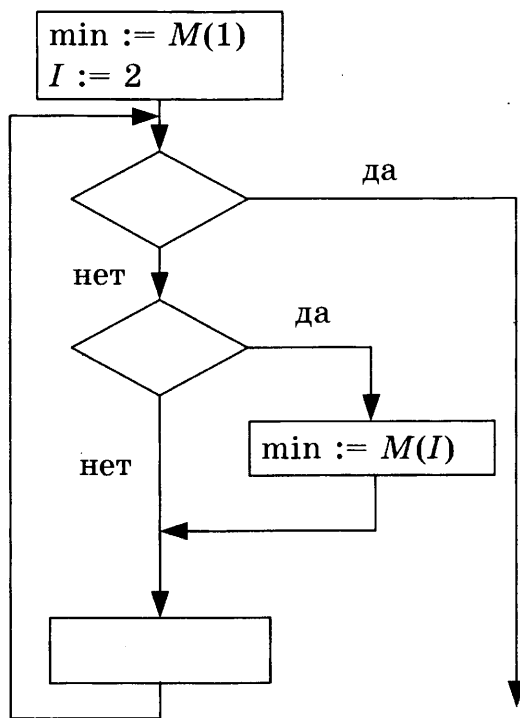
Итак, ответ $c = 4$.

2-й способ

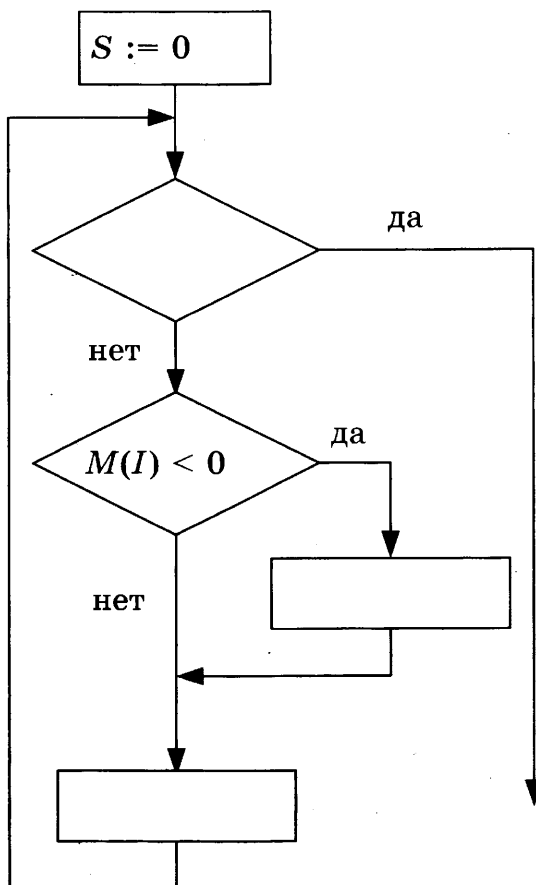
Внимательно проанализировав блок-схему, можно заметить, что она иллюстрирует последовательное удвоение переменной a , изначально равной единице, до тех пор, пока она не достигнет значения $16 = 2^4$. Отсюда $c = \log_2 16 = 4$.

5.5. Впишите во фрагмент блок-схемы пропущенные инструкции, необходимые для правильного решения поставленной задачи.

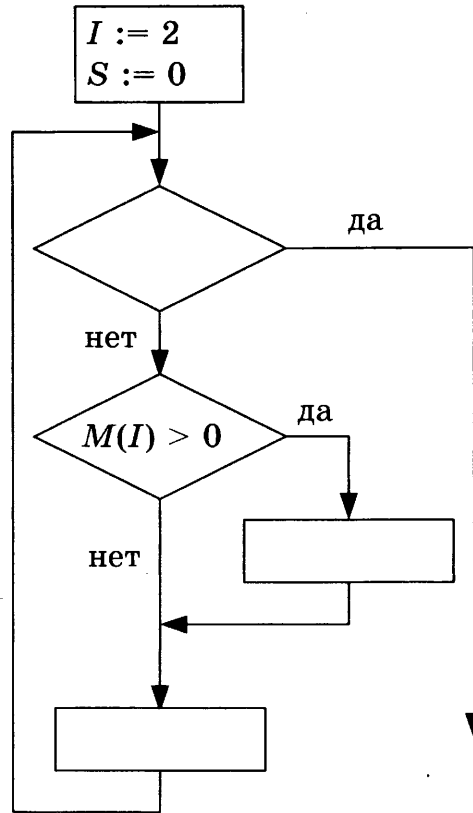
а) Найти минимальный элемент массива M из 24 элементов.



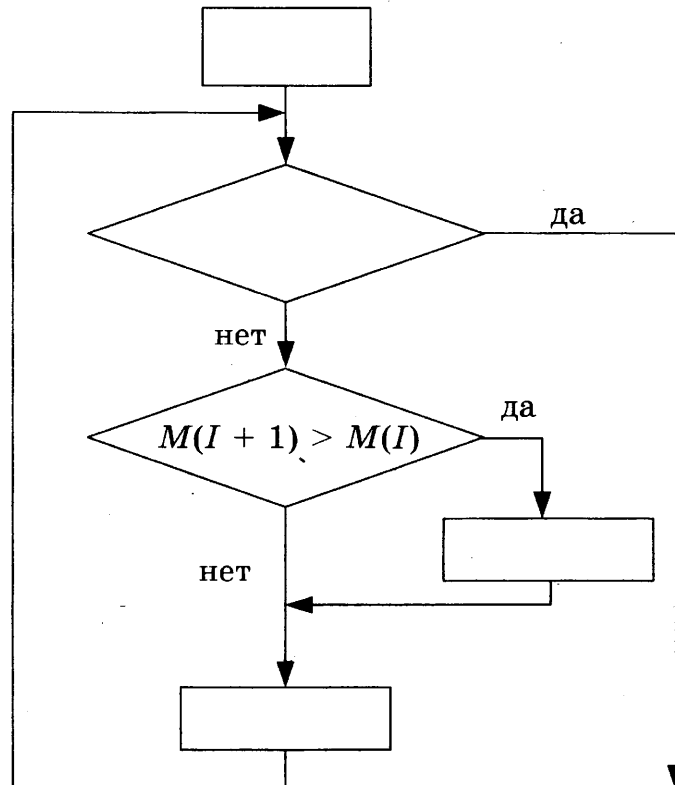
б) Найти сумму отрицательных элементов массива M из 24 элементов.



- в) Найти сумму положительных элементов массива M из 22 элементов, имеющих ч номера (нумерация элементов массива начинается с единицы).

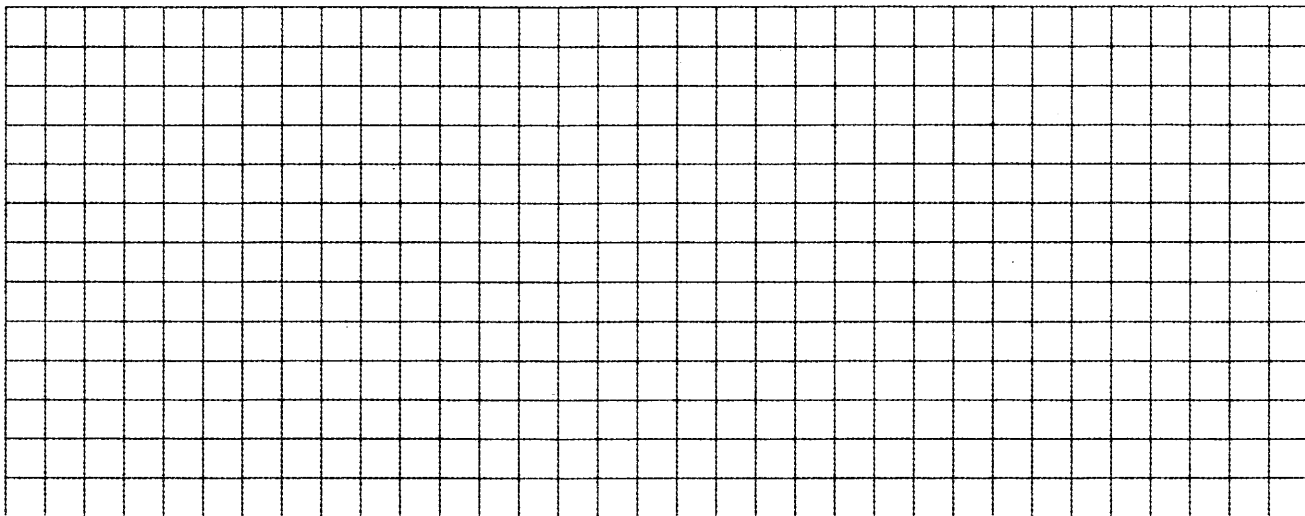


- г) В массиве M 45 элементов. Найти, сколько подряд идущих элементов этого массива начиная с первого, образуют возрастающую последовательность.

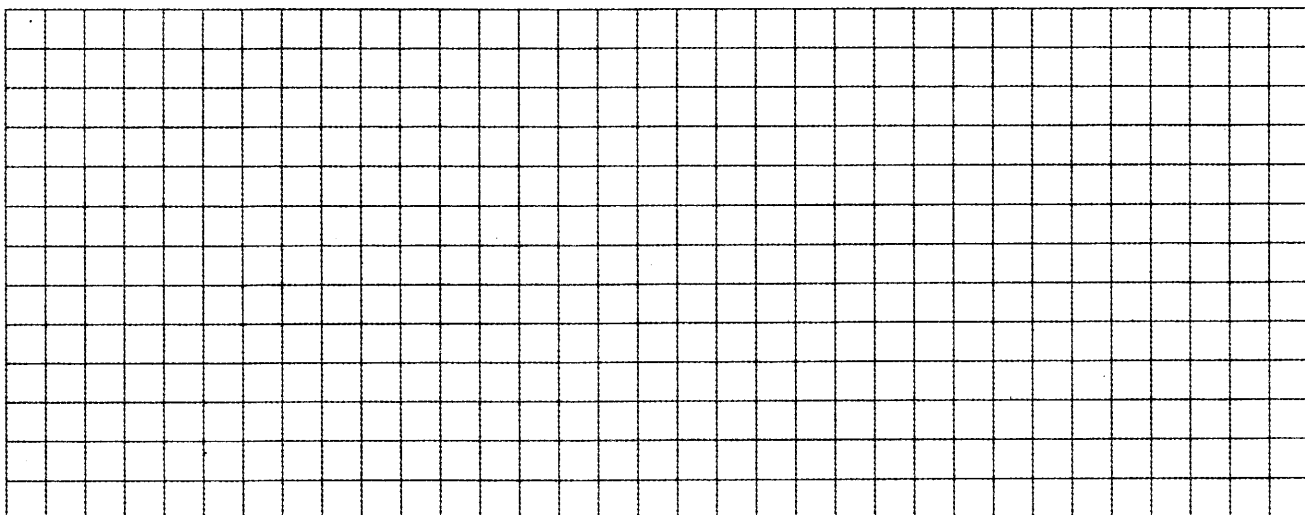


5.6. Составить и записать на естественном языке алгоритм, выполняющий:

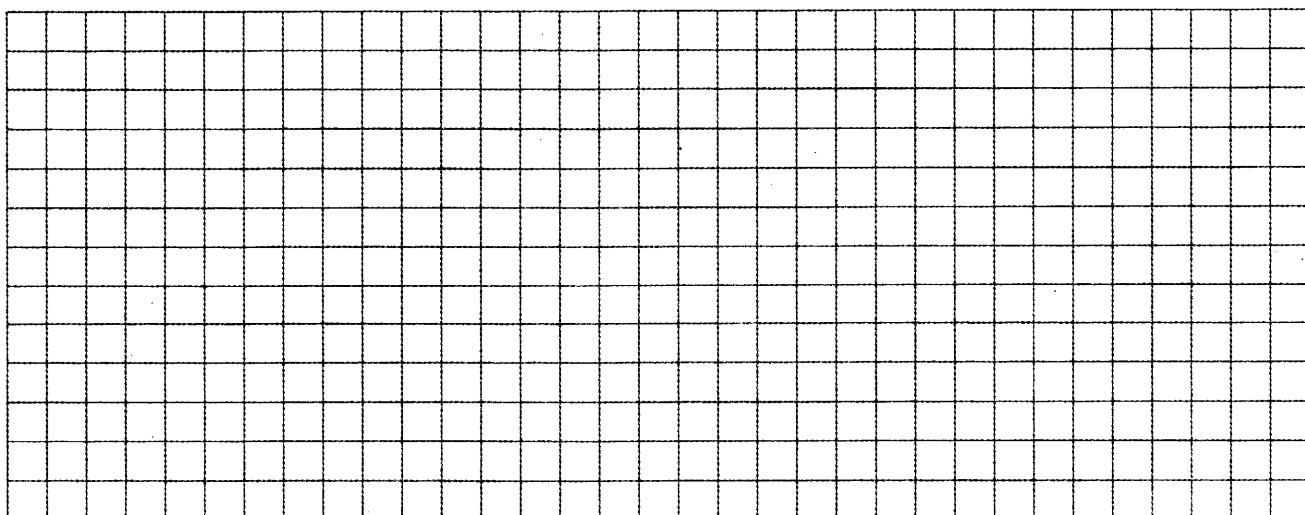
а) решение уравнения $ax = b$, где a и b — числовые исходные данные;



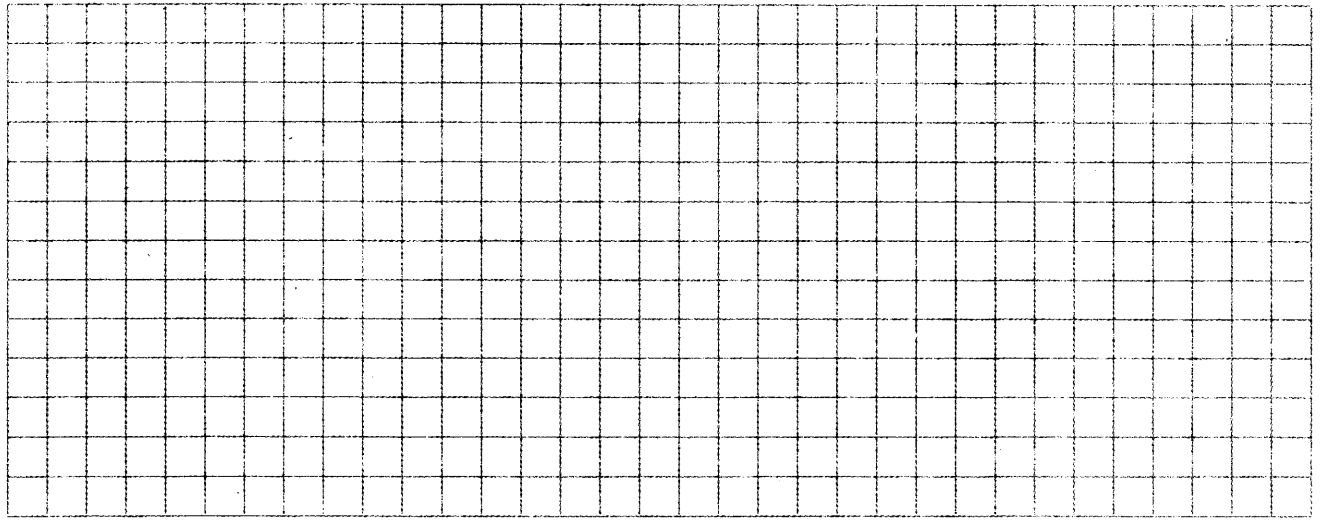
б) поиск количества минимальных элементов в заданном целочисленном массиве из 50 элементов;



в) поиск наибольшего общего делителя двух натуральных чисел;

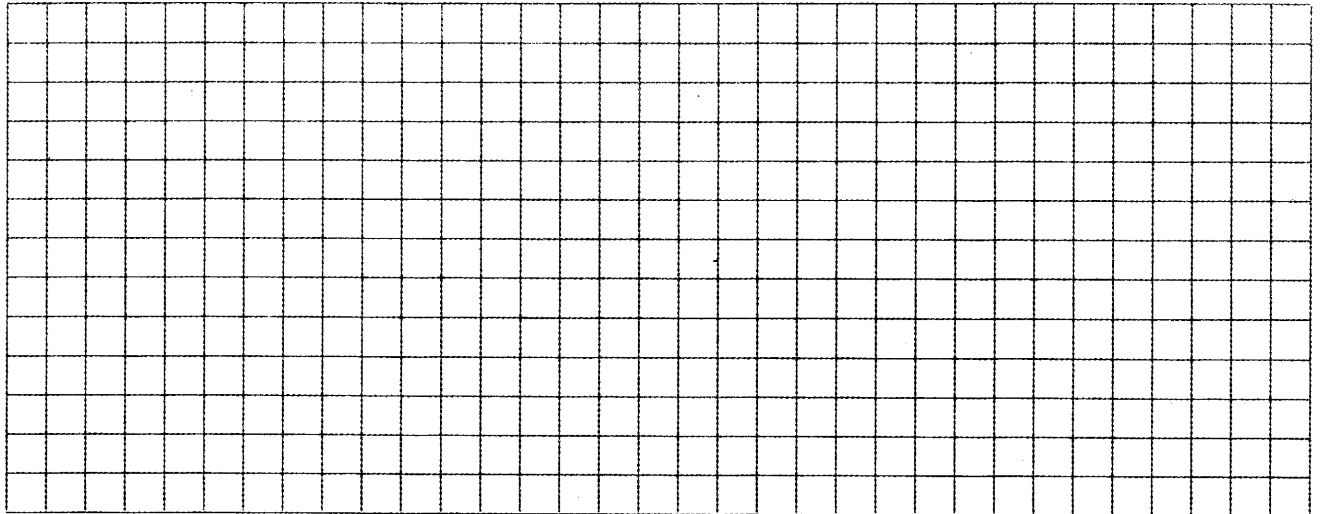


г) поиск количества простых чисел в заданном целочисленном массиве из 50 элементов.

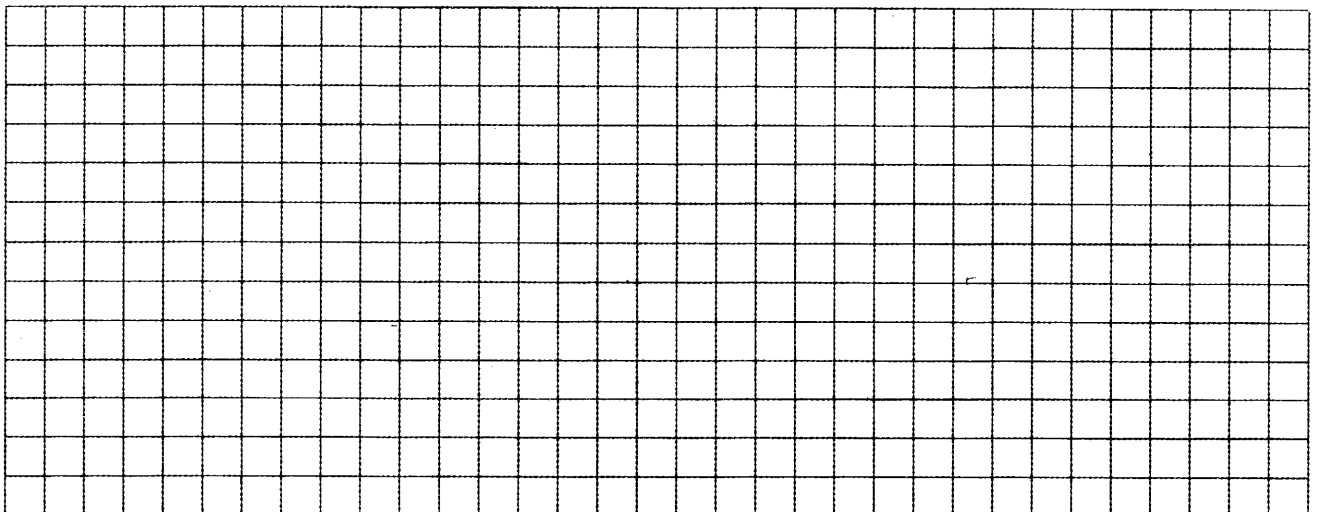


5.7. Составить и записать в виде блок-схемы алгоритм, выполняющий:

а) решение уравнения $ax = b$, где a и b — числовые исходные данные;



б) поиск количества минимальных элементов в заданном целочисленном массиве из 50 элементов;



в) Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический
<pre> j := A(1) FOR i = 2 TO n IF j < A(i) THEN j:=A(i) NEXT I </pre>	<pre> j:=A[1]; for i:=2 to n do if j<A[i] then j:=A[i]; </pre>	<pre> j:=A[1] нц для i от 2 до n если j<A[i] то j:=A[i] все кц </pre>

Как зависит итоговое значение переменной j от значений массива A ?

Ответ: _____

г) Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический
<pre> j = 0 FOR i = 1 TO n IF A(i)<0 THEN j=j+1 NEXT I </pre>	<pre> j:=0; for i:=1 to n do if A[i]<0 then j:=j+1; </pre>	<pre> j:=0 нц для i от 1 до n если A[i]<0 то j:=j+1 все кц </pre>

Как зависит итоговое значение переменной j от значений массива A ?

Ответ: _____

5.10.

а) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический
<pre> FOR i=1 TO n FOR k=1 TO n IF i>k THEN A(i,k) = 1 ELSE A(i,k) = 0 END IF NEXT k NEXT n </pre>	<pre> for i:=1 to n do for k:=1 to n do if i>k then A[i,k]:=1 else A[i,k]:=0 </pre>	<pre> нц для i от 1 до n нц для k от 1 до n если i>k то A[i,k]:=1 иначе A[i,k]:=0 все кц кц </pre>

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A , в зависимости от n .

Ответ: _____

- б) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический
<pre>FOR i=1 TO n FOR k=1 TO n IF i=k THEN A(i,k) = 1 ELSE A(i,k) = -1 END IF NEXT k NEXT n</pre>	<pre>for i:=1 to n do for k:=1 to n do if i=k then A[i,k]:=1 else A[i,k]:=-1</pre>	<pre><u>нц</u> <u>для</u> <u>i</u> <u>от</u> 1 <u>до</u> n <u>нц</u> <u>для</u> <u>k</u> <u>от</u> 1 <u>до</u> n <u>если</u> <u>i=k</u> <u>то</u> A[i,k]:=1 <u>иначе</u> A[i,k]:=-1 <u>все</u> <u>кц</u> <u>кц</u></pre>

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A в зависимости от n .

Ответ: _____.

- в) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический
<pre>FOR i=1 TO n FOR k=1 TO n IF i>k THEN A(i,k) = 1 ELSE A(i,k) = -1 END IF NEXT k NEXT n</pre>	<pre>for i:=1 to n do for k:=1 to n do if i>k then A[i,k]:=1 else A[i,k]:=-1</pre>	<pre><u>нц</u> <u>для</u> <u>i</u> <u>от</u> 1 <u>до</u> n <u>нц</u> <u>для</u> <u>k</u> <u>от</u> 1 <u>до</u> n <u>если</u> <u>i>k</u> <u>то</u> A[i,k]:=1 <u>иначе</u> A[i,k]:=-1 <u>все</u> <u>кц</u> <u>кц</u></pre>

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A в зависимости от n .

Ответ: _____.

- г) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

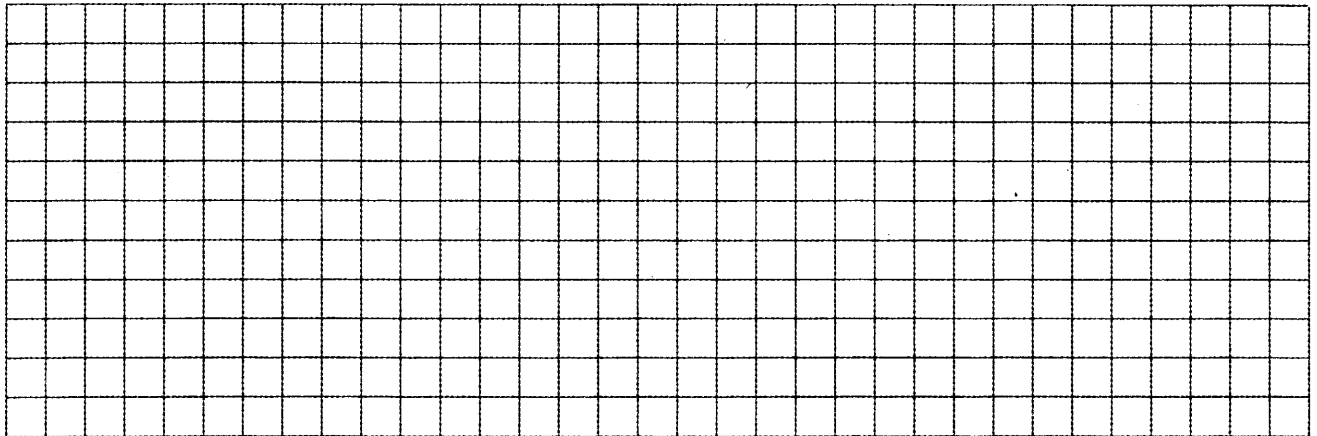
Бейсик	Паскаль	Алгоритмический
<pre>FOR i=1 TO n FOR k=1 TO n IF i>k THEN A(i,k) = i ELSE A(i,k) = -k END IF NEXT k NEXT n</pre>	<pre>for i:=1 to n do for k:=1 to n do if i>k then A[i,k]:=i else A[i,k]:=-k</pre>	<pre><u>нц</u> <u>для</u> <u>i</u> <u>от</u> 1 <u>до</u> n <u>нц</u> <u>для</u> <u>k</u> <u>от</u> 1 <u>до</u> n <u>если</u> <u>i>k</u> <u>то</u> A[i,k]:=i <u>иначе</u> A[i,k]:=-k <u>все</u> <u>кц</u> <u>кц</u></pre>

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A в зависимости от n .

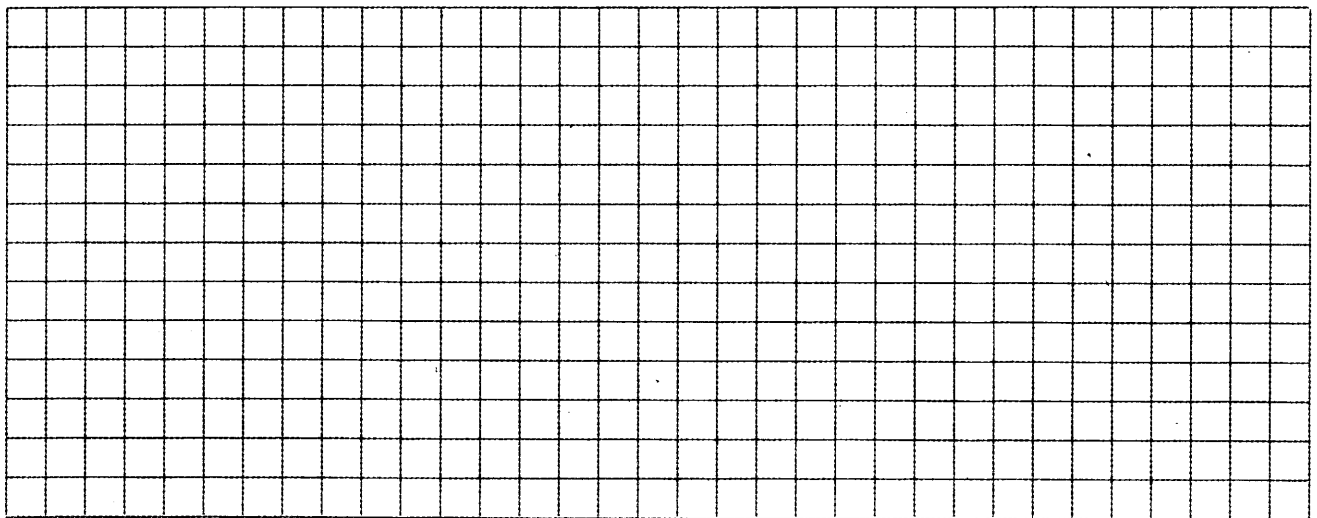
Ответ: _____.

5.11. Составить и записать в виде программы на любом языке программирования алгоритм, выполняющий:

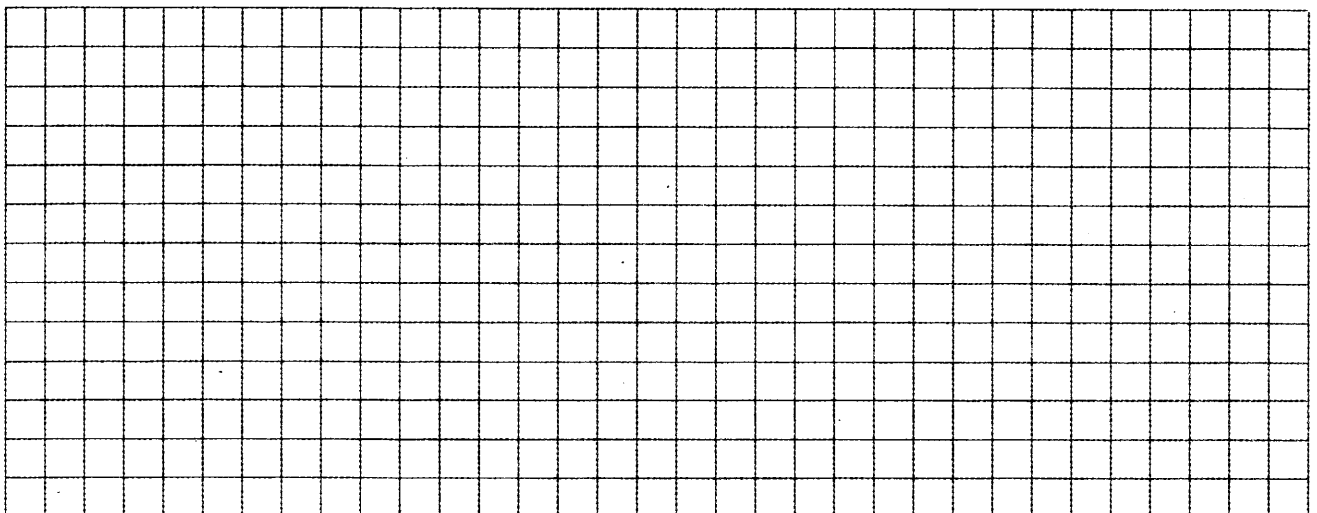
а) решение уравнения $a|x| = b$, где a и b — исходные числовые данные;



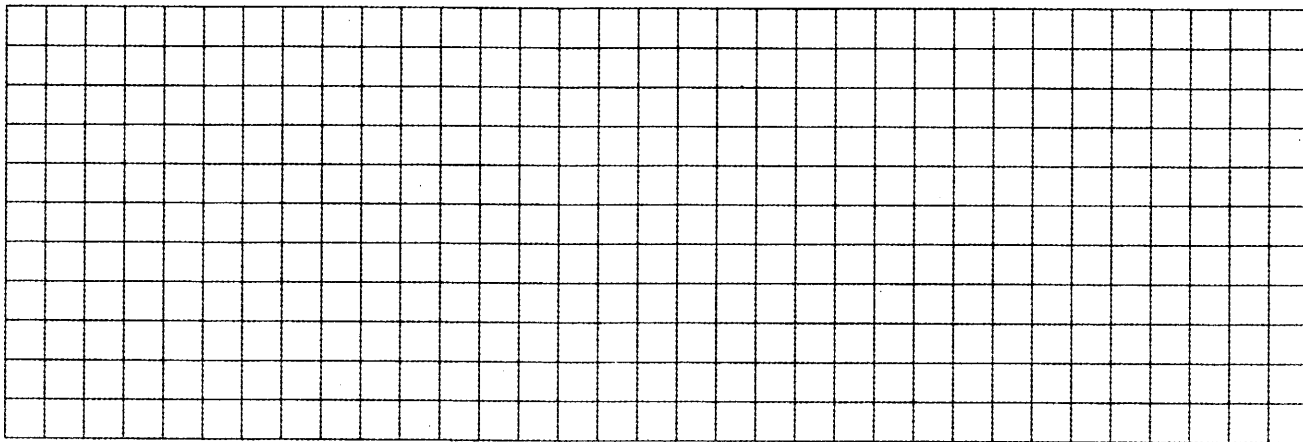
б) поиск количества минимальных элементов в заданном целочисленном массиве из 50 элементов;



в) поиск наибольшего общего делителя двух натуральных чисел;



г) поиск количества простых чисел в заданном целочисленном массиве из 50 элементов.



Важные рекомендации

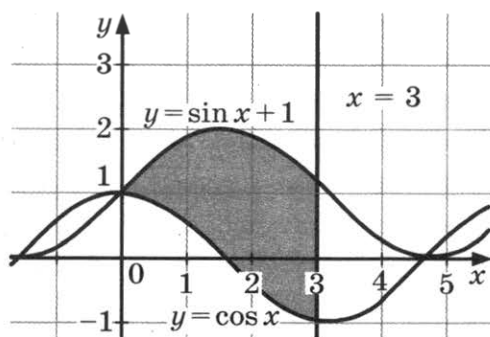
Практика показывает, что наибольшие затруднения при выполнении заданий по теме «Алгоритмизация и программирование» вызывает составление программ и алгоритмов. Данное пособие не является учебником программирования, поэтому мы приведем лишь общие рекомендации, не зависящие от языка, на котором записываются программы:

- Перед составлением программы надо тщательно проанализировать условие задания, уяснить постановку задачи, формат входных и выходных данных.
- Приступать к написанию программы можно лишь тогда, когда детально продуман алгоритм решения задачи, не следует начинать писать программу, не представляя алгоритмического решения. Для простой задачи алгоритмическое решение можно держать и в голове, но лучше его зафиксировать на бумаге в виде пояснения или схемы.
- Для написания программы нужно пользоваться тем языком программирования, которым хорошо владеешь и который подходит для решения поставленной задачи. Не следует специально стремиться поразить экзаменаторов знанием экзотического или недавно появившегося языка программирования, поскольку дополнительные баллы за это не предусмотрены. Хорошая программа должна быть максимально простой и эффективной. Не нужно специально использовать сложные приемы или редко применяемые конструкции языка программирования, если в этом нет необходимости.
- Текст программы нужно писать аккуратно и разборчиво даже на черновике, структурированно, выделяя отступами уровень вложенности операторов, не скупясь на комментарии и пояснения. Отсутствие комментариев и пояснений вовсе не является признаком высокой квалификации программиста, скорее наоборот.
- При написании текста программы следует контролировать соответствие типов переменных и применяемых к ним операций, не допускать использования неопределенных (неинициализированных) значений переменных.
- Если текст программы получается, на ваш взгляд, очень сложным и запутанным, приостановите на время ее написание и подумайте, в чем причина этой сложности, нельзя ли найти более простое и изящное решение задачи.
- После того как программа написана, необходимо выполнить ее тестирование, то есть проверку правильности работы для различных исходных данных. Постарайтесь найти ошибку в своей программе, помните, что цель тестирования — поиск ошибок, а не демонстрация правильности программы. Гораздо лучше, если ошибку найдете и исправите вы сами, а не экзаменаторы при проверке.

- При тестировании следует подбирать набор тестовых данных так, чтобы все операторы, все «ветки» программы выполнились на совокупности тестов. Особое внимание следует уделить особым случаям исходных данных, пограничным, критическим точкам. Пусть, например, написана программа сортировки массива. Особыми здесь будут случаи, когда входной массив уже отсортирован в нужном порядке, когда он отсортирован в порядке, обратном требуемому, а также когда все элементы сортируемого массива равны. Следует выявить такие критические точки и убедиться, что программа правильно работает на этих тестах.
- Если в задании приведен пример входных данных, его тоже надо включить в состав тестов, но он не должен быть единственным тестом!
- При тестировании следует еще раз сверить структуру исходных и выходных данных с заданием, убедиться, что составленная программа точно соответствует условию задачи.
- При тестировании необходимо внимательно контролировать синтаксис программы (должен быть явно указан тип каждой переменной, если этого требует язык программирования, переменные должны использоваться в соответствии со своим типом, не должно быть неинициализированных и неиспользуемых переменных, должны быть правильно расставлены операторные скобки и разделители операторов и т.д.).
- Если после тестирования программы осталось достаточно времени, то можно поработать над повышением ее эффективности, подумать, нельзя ли упростить ее, уменьшить количество циклов или требуемой для ее работы памяти. В случае внесения изменений в готовую программу процедуру тестирования придется повторить заново.

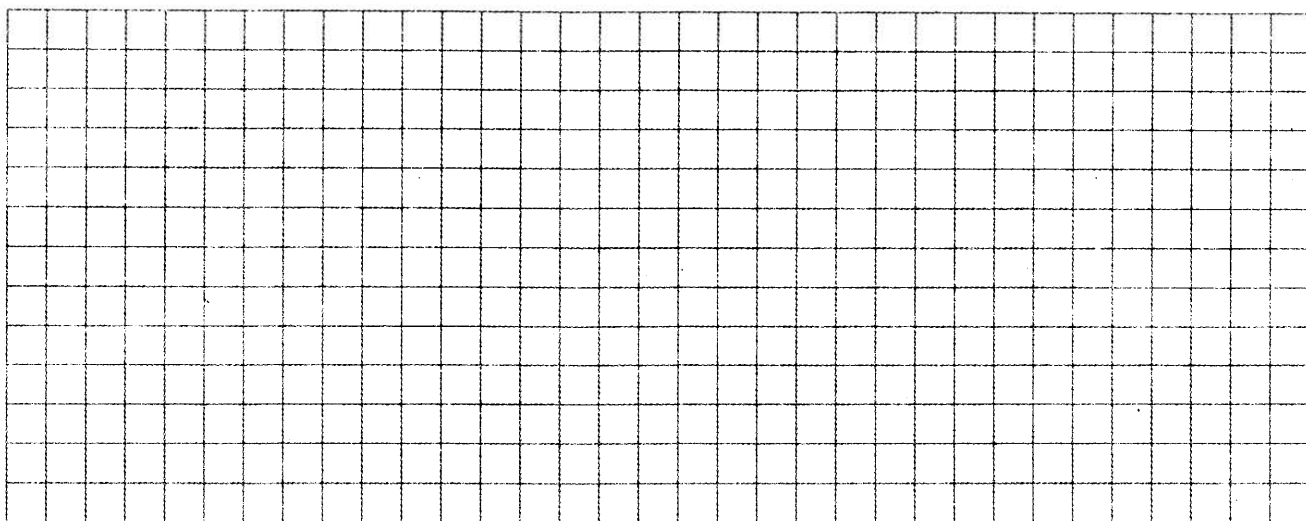
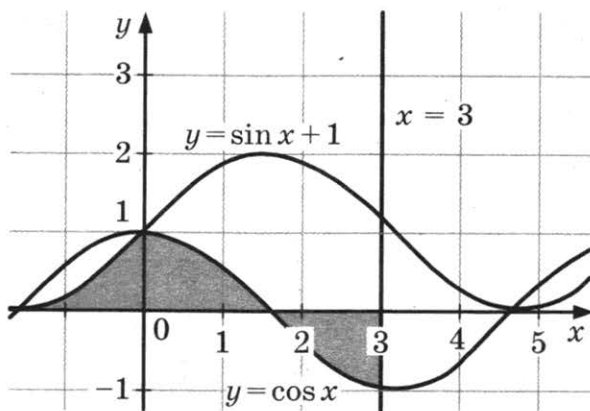
5.12.

- a) Найдите и исправьте ошибки в приведенной программе проверки принадлежности точки закрашенной области. Приведите пример, когда исходная программа работала неправильно.

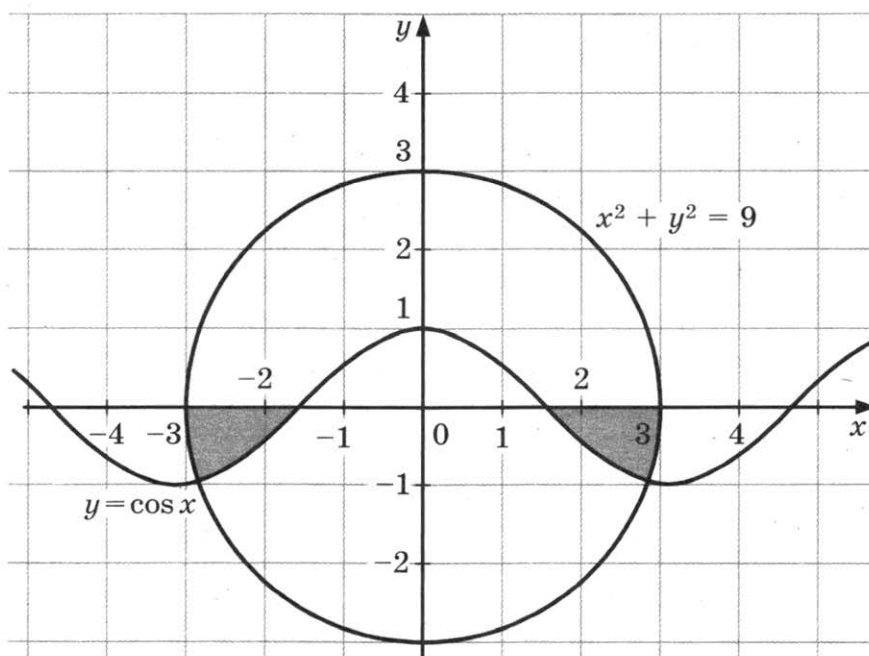


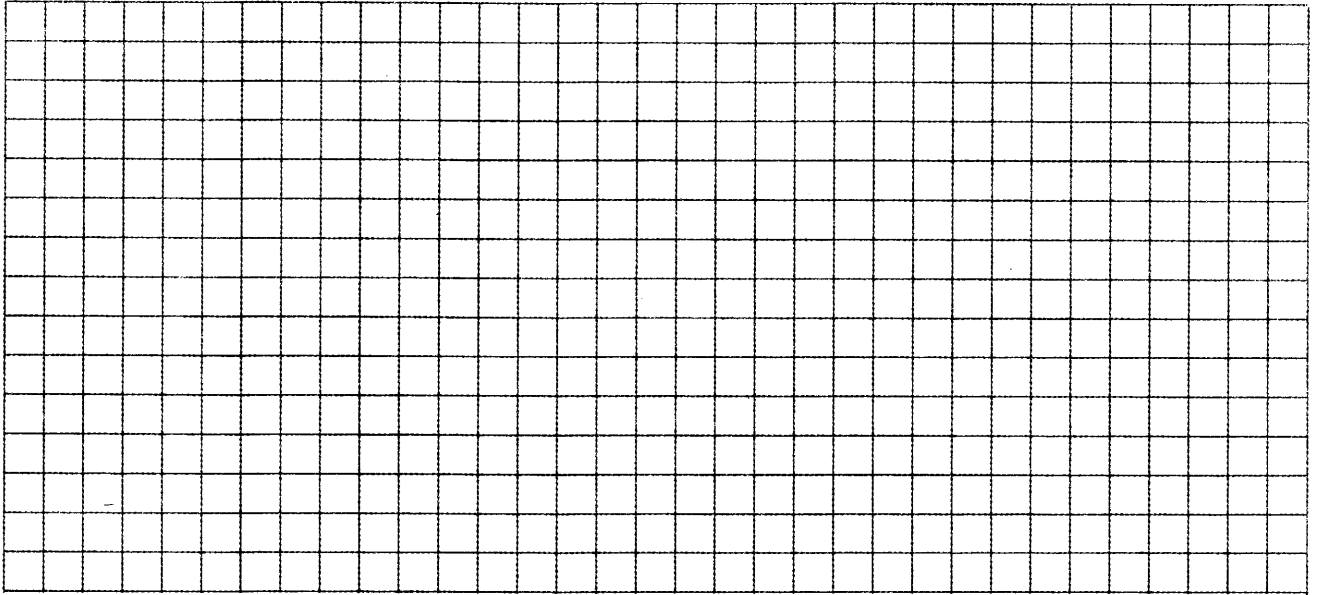
Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x,y: real; begin readln(x,y); if y<=sin(x)+1 then if y>=cos(x) then if x<=3 then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x, y IF y<=sin(x)+1 THEN IF y>=cos(x) THEN IF x<=3 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (y<=sin(x)+1) if (y>=cos(x)) if (x<=3) printf("принадлежит"); else printf("не принадлежит"); }</pre>

- б) Составьте программу проверки принадлежности точки закрашенной области.



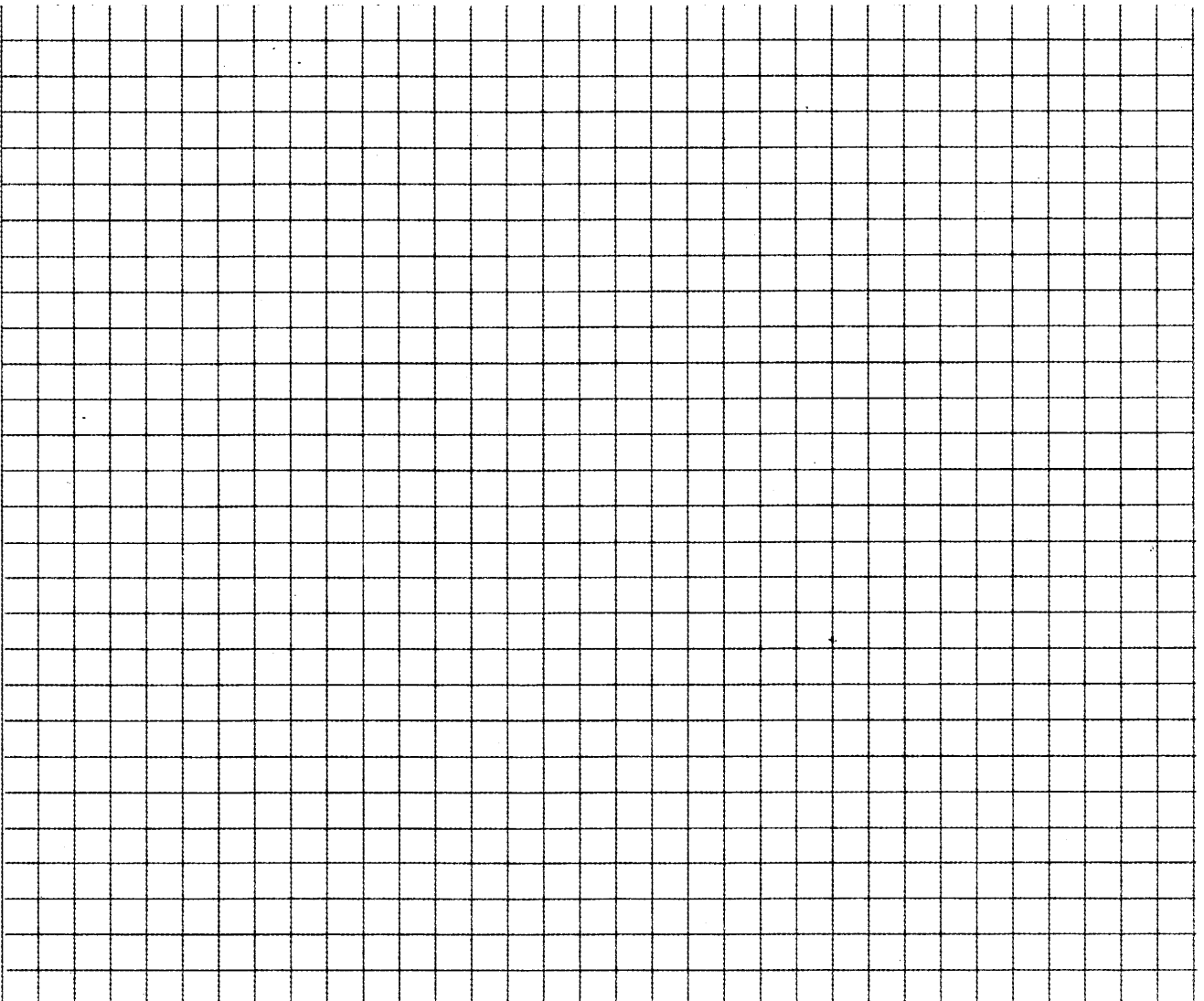
- в) Найдите и исправьте ошибки в приведенной программе проверки принадлежности точки закрашенной области. Приведите пример, когда исходная программа работала неправильно.



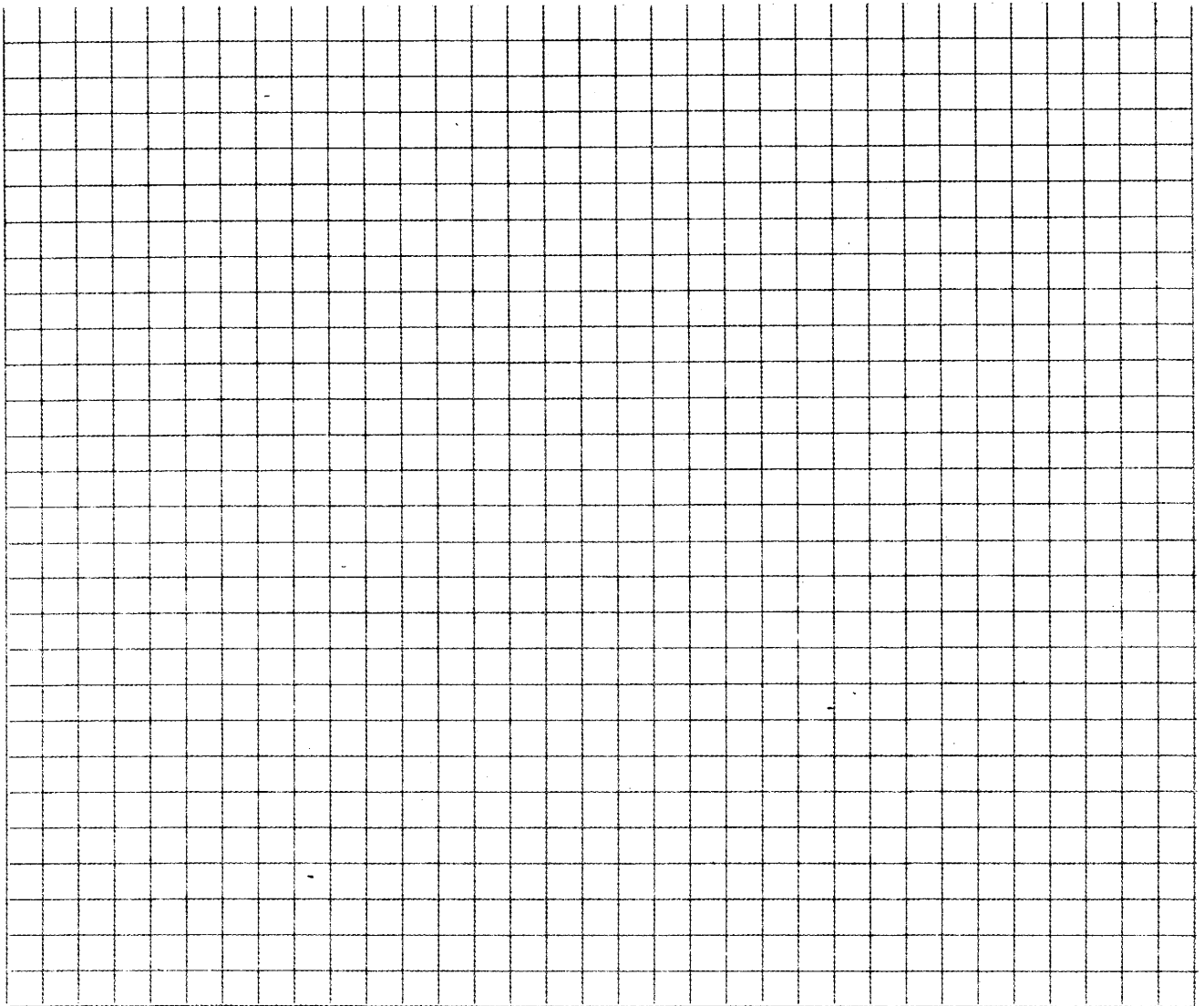


5.13.

- а) Составить программу, выводящую на экран все натуральные трехзначные числа, делящиеся без остатка на 17, сумма цифр которых равняется 11.

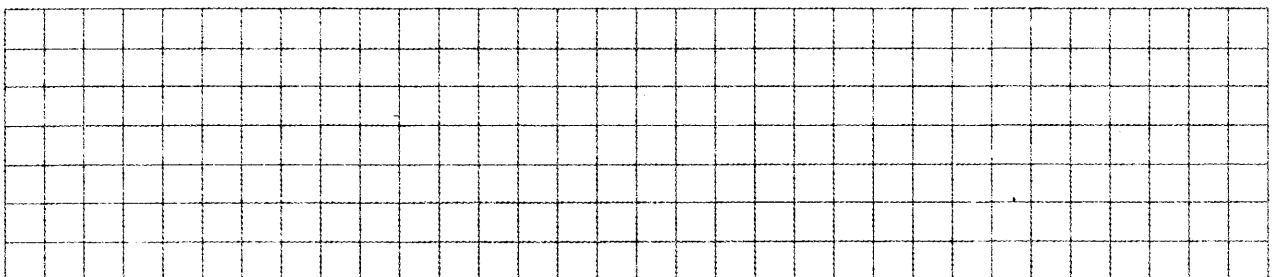


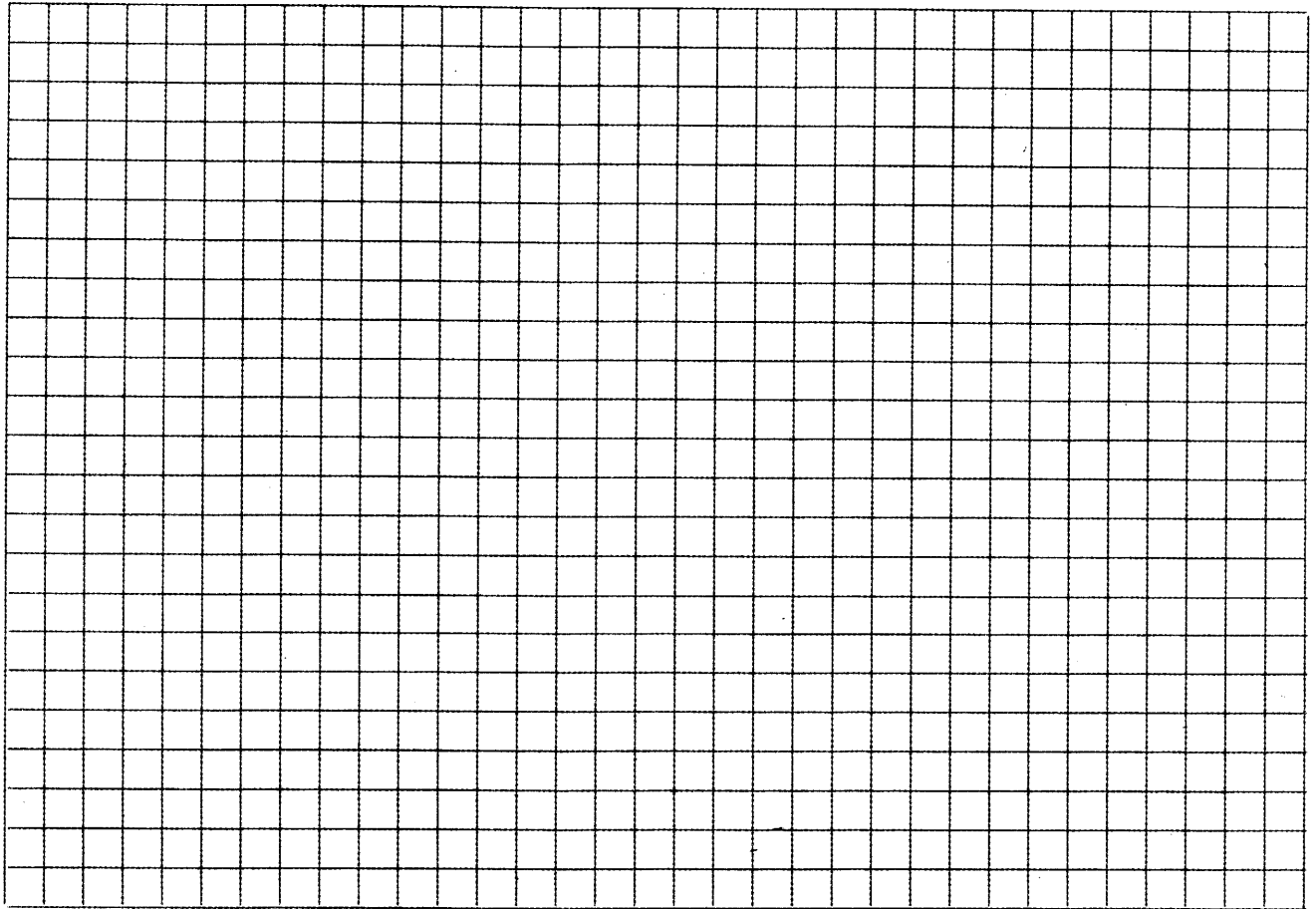
- б) Составить программу, выводящую на экран все натуральные трехзначные числа, делящиеся на 19 с остатком 3, сумма цифр которых равняется 12.



- в) На продовольственном складе имеется 10 000 наименований различных продуктов, информация о которых хранится в текстовом файле. Каждому продукту отведена 1 строка файла. В ней указаны числовой артикул продукта, название (не содержащее пробелов), категория продукта (молочный, мясной или рыбный) и его стоимость. Вышеперечисленные поля разделены одним пробелом.

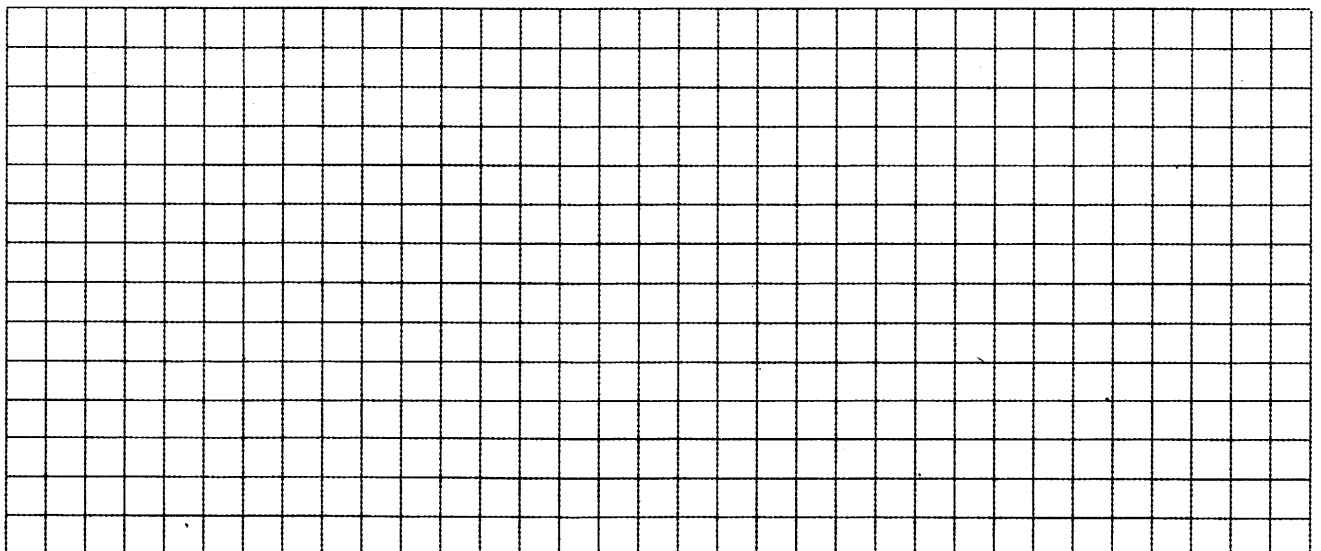
Написать программу, загружающую в память компьютера информацию о продуктах из текстового файла и выдающую на экран монитора наименования и артикулы всех рыбных продуктов, стоимость которых отличается от максимальной по данной категории не более чем на 10%.

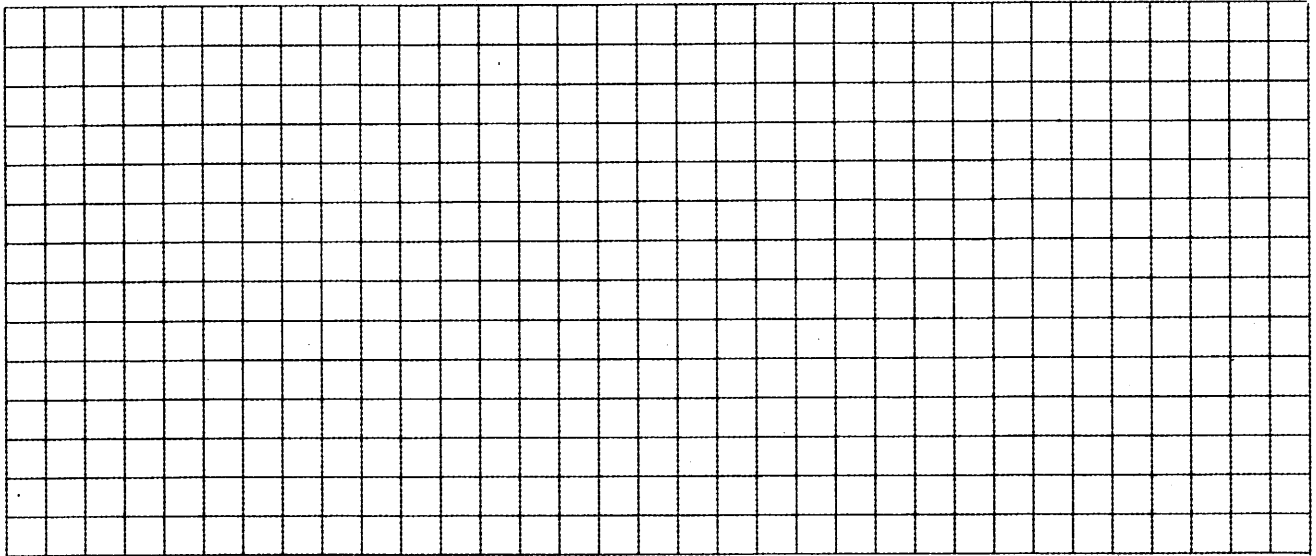




- г) На предприятии работает 1000 сотрудников, информация о которых хранится в текстовом файле. Каждому сотруднику отведена 1 строка файла. В ней указаны целочисленный индивидуальный табельный номер, фамилия сотрудника (не содержащая пробелов), стаж (целое число лет) и зарплата. Вышеперечисленные поля разделены одним пробелом.

Написать программу, загружающую в память компьютера информацию о сотрудниках из текстового файла и выдающую на экран фамилии и табельные номера всех сотрудников, проработавших не менее 8 лет, зарплата которых меньше средней зарплаты по предприятию более чем на 25%.





Задания в формате ЕГЭ

5.14.

- а) Для составления цепочек используются бусины, помеченные буквами I, M, N, O, P. Цепочка формируется по следующему правилу: в середине цепочки стоит одна из бусин I, M, O. На третьем месте — бусина, помеченная любой гласной, если первая бусина — согласной, или любой согласной, если первая — гласной. На первом месте — одна из бусин I, O, P, не стоящая в цепочке посередине.

Какая из перечисленных цепочек создана по этому правилу?

- 1) OPN 2) P I I 3) POM 4) I I O

- б) Для составления цепочек используются бусины, помеченные буквами A, M, N, O, P. Цепочка формируется по следующему правилу: в середине цепочки стоит одна из бусин M, O. На третьем месте — бусина, помеченная любой гласной, если первая бусина — согласной, или любой согласной, если первая — гласной. На первом месте — одна из бусин N, O, P, не стоящая в цепочке посередине.

Какая из перечисленных цепочек создана по этому правилу?

- 1) NMA 2) NOM 3) OON 4) NAO

- в) Для составления цепочек используются бусины, помеченные буквами A, Б, Г, Е, О. Цепочка формируется по следующему правилу: бусины с гласной и согласной буквой не могут стоять рядом. В середине цепочки может стоять одна из бусин Б или Е, не стоящая на первом месте. На третьем месте — бусина, помеченная любой гласной.

Какая из перечисленных цепочек создана по этому правилу?

- 1) AGE 2) AEA 3) GBA 4) ABГ

- г) Дешифровщику необходимо восстановить поврежденный фрагмент сообщения, состоящий из 4 символов. Имеется достоверная информация, что использовано не более пяти букв (A, Б, Е, В, Г), причем на втором месте стоит один из символов A, Г, Е, В. На первом — одна из букв Б, В, Г, Е, которой нет на третьем месте. На третьем месте — одна из букв Е, А, Б, В, не стоящая в слове на втором месте. На четвертом месте — одна из букв A, Б, В, Г, которой не было на первом месте.

Ответ: _____

- б) Запишите порядок команд в программе получения из числа 4 числа 31 для исполнителя Утроитель, содержащей не более 5 команд, указывая лишь номера команд.

Ответ: _____

- в) У исполнителя Квадратор две команды, которым присвоены номера:

1. Прибавь 1

2. Возведи в квадрат

Первая из них увеличивает число на экране на 1, вторая — возводит его в квадрат.

Запишите порядок команд в программе получения из числа 6 числа 66 для исполнителя Квадратор, содержащей не более 5 команд, указывая лишь номера команд.

Ответ: _____

- г) Запишите порядок команд в программе получения из числа 2 числа 50 для исполнителя Квадратор, содержащей не более 6 команд, указывая лишь номера команд.

Ответ: _____

- 5.18.** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
--------------	-------------	--------------	---------------

При выполнении этой команды РОБОТ перемещается на соответствующую клетку. Команды проверки истинности условия на наличие стены у той клетки, где он находится:

сверху свободно	снизу свободно	слева свободно	справа свободно
------------------------	-----------------------	-----------------------	------------------------

Если РОБОТ начнет движение в сторону стены, то он разрушится.

- а) Сколько клеток данного лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО

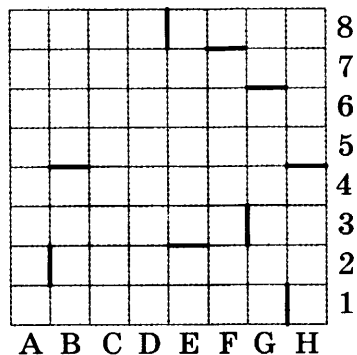
ПОКА справа свободно
ДЕЛАТЬ вправо

ПОКА снизу свободно
ДЕЛАТЬ вниз

ПОКА слева свободно
ДЕЛАТЬ влево

ПОКА сверху свободно
ДЕЛАТЬ вверх

КОНЕЦ



В ответе запишите число — количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).

Например, нижний левый угол лабиринта имеет адрес A1.

Ответ: _____.

- б) Сколько клеток данного лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО

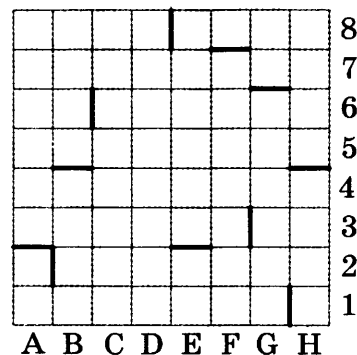
ПОКА сверху свободно
ДЕЛАТЬ вверх

ПОКА слева свободно
ДЕЛАТЬ влево

ПОКА снизу свободно
ДЕЛАТЬ вниз

ПОКА справа свободно
ДЕЛАТЬ вправо

КОНЕЦ



В ответе запишите число — количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).

Например, нижний левый угол лабиринта имеет адрес A1.

Ответ: _____.

- в) Сколько клеток данного лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО

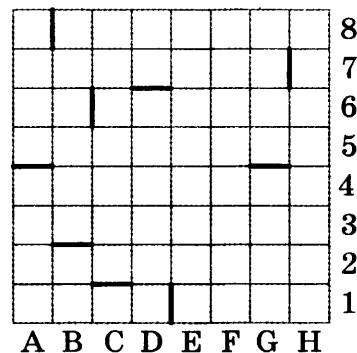
ПОКА сверху свободно
ДЕЛАТЬ вверх

ПОКА слева свободно
ДЕЛАТЬ влево

ПОКА снизу свободно
ДЕЛАТЬ вниз

ПОКА справа свободно
ДЕЛАТЬ вправо

КОНЕЦ



В ответе запишите число — количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).

Например, нижний левый угол лабиринта имеет адрес A1.

Ответ: _____.

- г) Сколько клеток данного лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО

ПОКА справа свободно

ДЕЛАТЬ вправо

ПОКА сверху свободно

ДЕЛАТЬ вверх

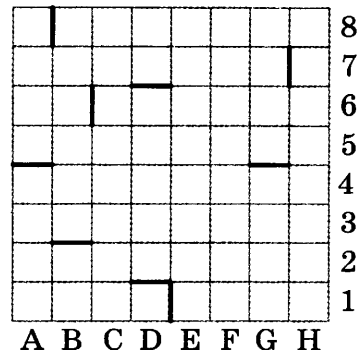
ПОКА слева свободно

ДЕЛАТЬ влево

ПОКА снизу свободно

ДЕЛАТЬ вниз

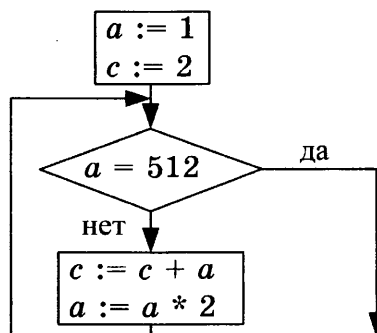
КОНЕЦ



Ответ: _____.

5.19. Определите значение переменной c после выполнения фрагмента алгоритма:

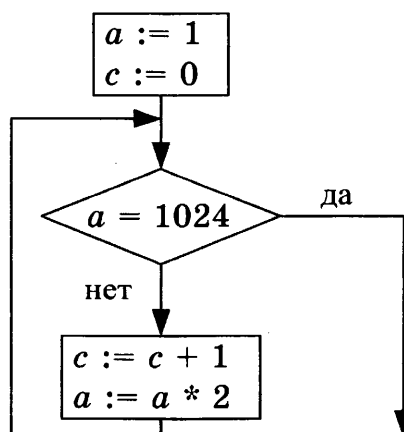
а)



Примечание: знаком := обозначена операция присваивания.
знаком * обозначена операция умножения

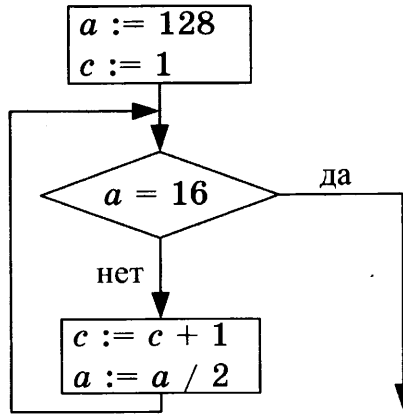
Ответ: _____.

б)



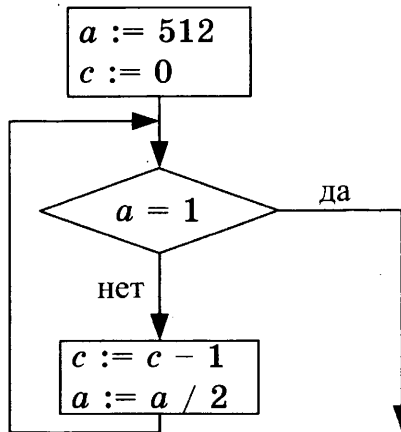
Ответ: _____.

в)



Ответ: _____

г)



Ответ: _____

5.20. Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

а)

Бейсик	Паскаль	Алгоритмический
$a=3+2 * 4$ $b=(a \text{ MOD } 10)+24$ $a=(b \backslash 10)+1$ \ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a:=3+2 * 4;$ $b:=(a \text{ mod } 10)+24;$ $a:=(b \text{ div } 10)+1$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a:=3+2 * 4$ $b:=\text{mod}(a, 10)+24$ $a:=\text{div}(b, 10)+1$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 3, b = 24$ 2) $a = 3, b = 25$ 3) $a = 4, b = 25$ 4) $a = 4, b = 35$

б)

Бейсик	Паскаль	Алгоритмический
$a=(3+2) * 4$ $b=(a \text{ MOD } 5)+11$ $a=(b \backslash 3)+1$	$a:=(3+2) * 4;$ $b:=(a \text{ mod } 5)+11;$ $a:=(b \text{ div } 3)+1$	$a:=(3+2) * 4$ $b:=\text{mod}(a, 5)+11$ $a:=\text{div}(b, 3)+1$

- 1) $a = 4, b = 11$ 2) $a = 4, b = 12$ 3) $a = 5, b = 11$ 4) $a = 6, b = 15$

в)	Бейсик	Паскаль	Алгоритмический
	<pre>a=(3-2) * 4 b=(a MOD 5)+12 a=(b\3)+1</pre>	<pre>a:=(3-2) * 4; b:=(a mod 5)+12; a:=(b div 3)+1</pre>	<pre>a:=(3-2) * 4 b:=mod(a,5)+12 a:=div(b, 3)+1</pre>

- 1) $a = 2, b = 12$ 2) $a = 2, b = 16$ 3) $a = 3, b = 12$ 4) $a = 6, b = 16$

г)	Бейсик	Паскаль	Алгоритмический
	<pre>a=24-2 * 4 b=(a MOD 6)+12 a=(b\7)+1</pre>	<pre>a:=24-2 * 4; b:=(a mod 6)+12; a:=(b div 7)+1</pre>	<pre>a:=24-2 * 4 b:=mod(a,6)+12 a:=div(b, 7)+1</pre>

- 1) $a = 2, b = 22$ 2) $a = 4, b = 22$ 3) $a = 3, b = 16$ 4) $a = 6, b = 16$

5.21.

а) Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический
<pre>j=1 FOR i=1 TO n IF A(i) < A(j) THEN j = i NEXT i s=j</pre>	<pre>j:=1; for i:=1 to n do begin if A[i]<A[j] then j:=i end s:=j</pre>	<pre>j:=1 нц для i от 1 до n если A[i]<A[j] то j:=i все кц s:=j</pre>

Чему будет равно значение переменной s после выполнения данного алгоритма при любых значениях элементов массива A ?

- 1) Индексу минимального элемента в массиве A (первому из них, если минимальных элементов несколько)
- 2) Индексу минимального элемента в массиве A (последнему из них, если минимальных элементов несколько)
- 3) Индексу максимального элемента в массиве A (последнему из них, если максимальных элементов несколько)
- 4) Индексу максимального элемента в массиве A (последнему из них, если максимальных элементов несколько)

б) Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический
<pre>j = 1 FOR i = 1 TO n IF A(i) = A(j) THEN j = i NEXT i s=j</pre>	<pre>j:=1; for i:=1 to n do begin if A[i]=A[j] then j:=i end; s:=j</pre>	<pre>j:=1 нц для i от 1 до n если A[i]=A[j] то j:=i все кц s:=j</pre>

Чему будет равно значение переменной s после выполнения данного алгоритма при любых значениях элементов массива A ?

- 1) Индексу элемента, равного первому и имеющему наибольший индекс
- 2) Индексу элемента, равного последнему и имеющему наименьший индекс
- 3) Значению n
- 4) Единице

в) Дан фрагмент программы, обрабатывающей двумерный массив A размера $n \times n$.

Бейсик	Паскаль	Алгоритмический
<pre>k = 1 FOR i = 1 TO n c = A(i, k+2) A(i, k+2) = A(i, k) A(i, k) = c NEXT i</pre>	<pre>k:=1; for i:=1 to n do begin c:=A[i, k+2]; A[i, k+2]:=A[i, k]; A[i, k]:=c end</pre>	<pre>k:=1 нц для i от 1 до n c:=A[i, k+2] A[i, k+2]:=A[i, k] A[i, k]:=c кц</pre>

Представим массив в виде квадратной таблицы, в которой для элемента массива $A[i, j]$ величина i является номером строки, а величина j — номером столбца, в котором расположен элемент. Тогда данный алгоритм меняет местами:

- 1) два столбца в таблице
- 2) две строки в таблице
- 3) элементы диагонали и k -й строки таблицы
- 4) элементы диагонали и k -го столбца таблицы

г) Дан фрагмент программы, обрабатывающей двумерный массив A размера $n \times n$.

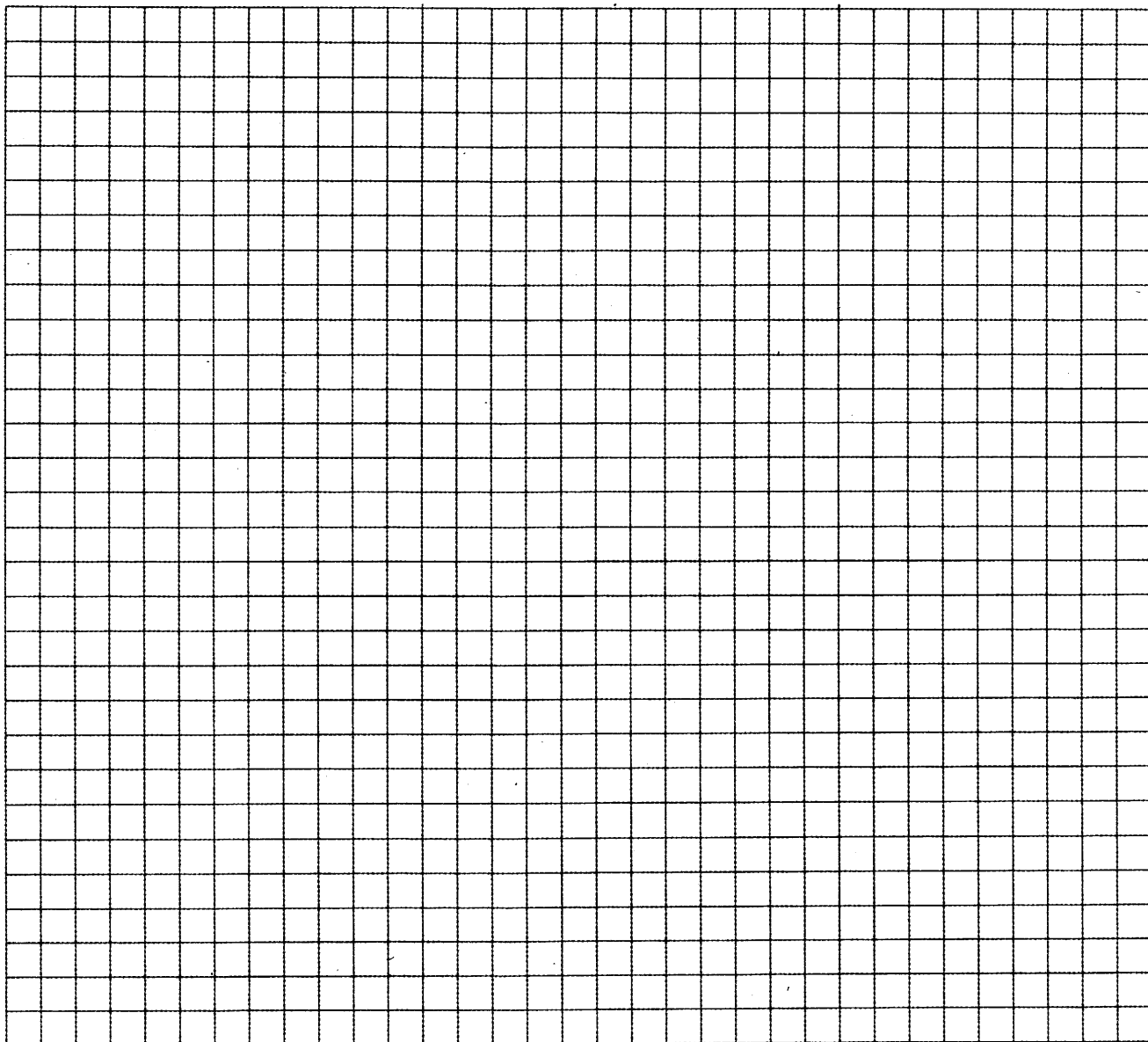
Бейсик	Паскаль	Алгоритмический
<pre>k = 3 FOR i = 1 TO n c = A(i, i) A(i, i) = A(i, k) A(i, k) = c NEXT i</pre>	<pre>k:=3; for i:=1 to n do begin c:=A[i, i]; A[i, i]:=A[i, k]; A[i, k]:=c end</pre>	<pre>k:=3 нц для i от 1 до n c:=A[i, i] A[i, i]:=A[i, k] A[i, k]:=c кц</pre>

Представим массив в виде квадратной таблицы, в которой для элемента массива $A[i, j]$ величина i является номером строки, а величина j — номером столбца, в котором расположен элемент. Тогда данный алгоритм меняет местами:

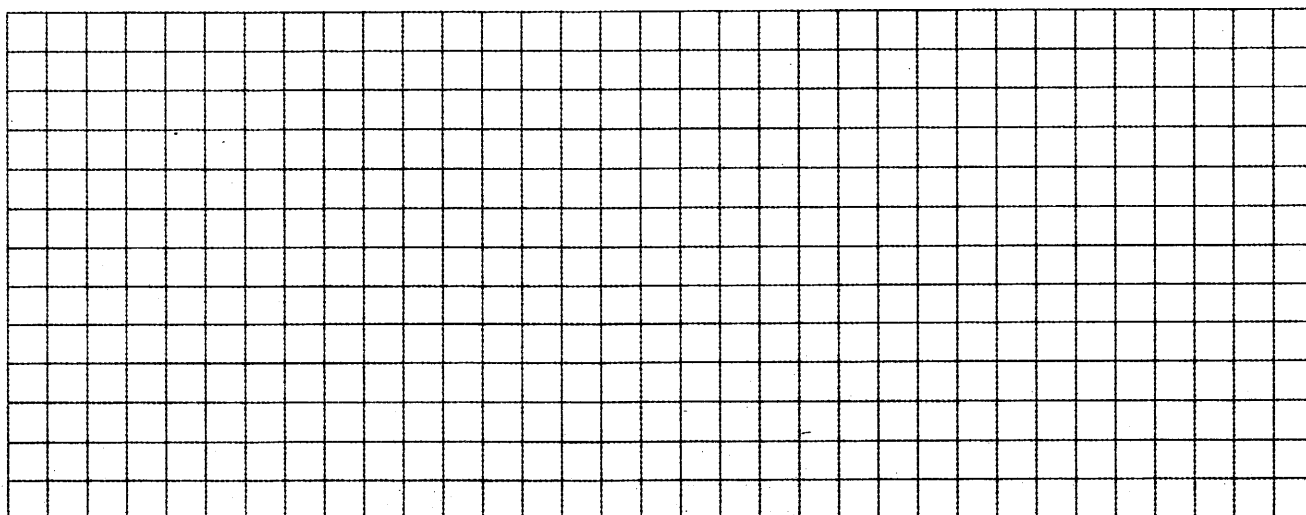
- 1) два столбца в таблице
- 2) две строки в таблице
- 3) элементы диагонали и k -й строки таблицы
- 4) элементы диагонали и k -го столбца таблицы

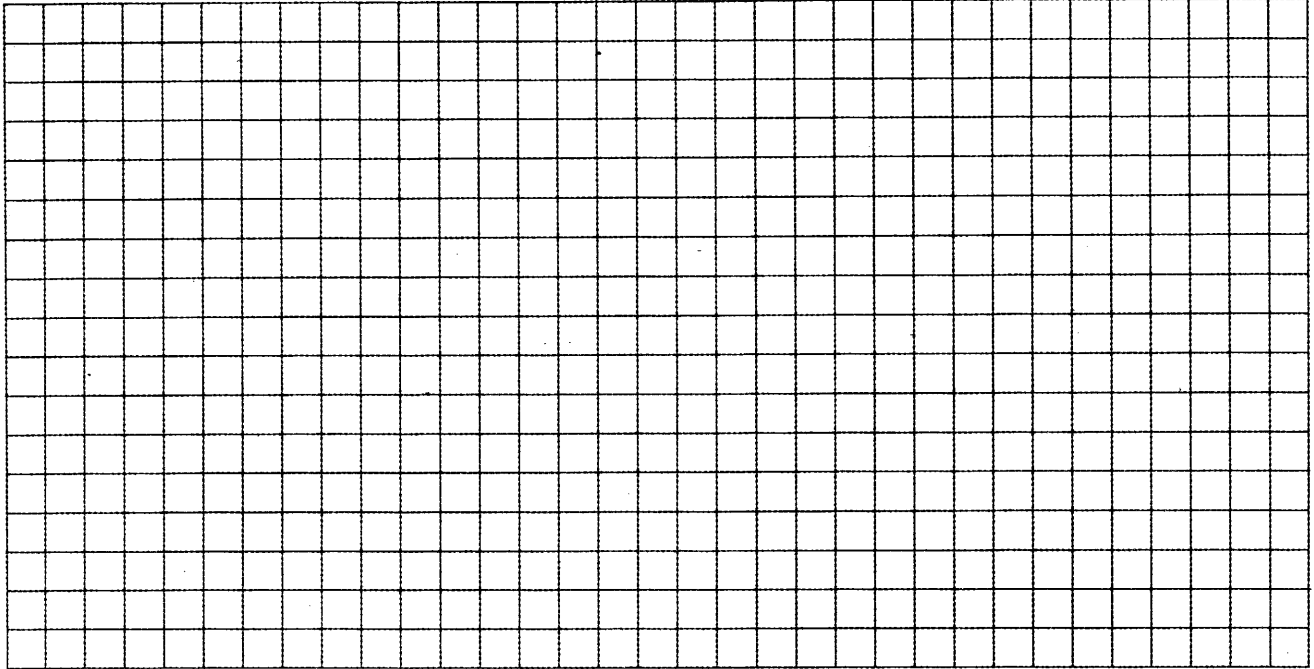
5.22. Составить и записать в виде программы на любом языке программирования или в виде блок-схемы или на естественном языке алгоритм, выполняющий:

- а) поиск минимального нечетного элемента в заданном целочисленном массиве из 50 элементов. Известно, что в массиве есть хотя бы один нечетный элемент;

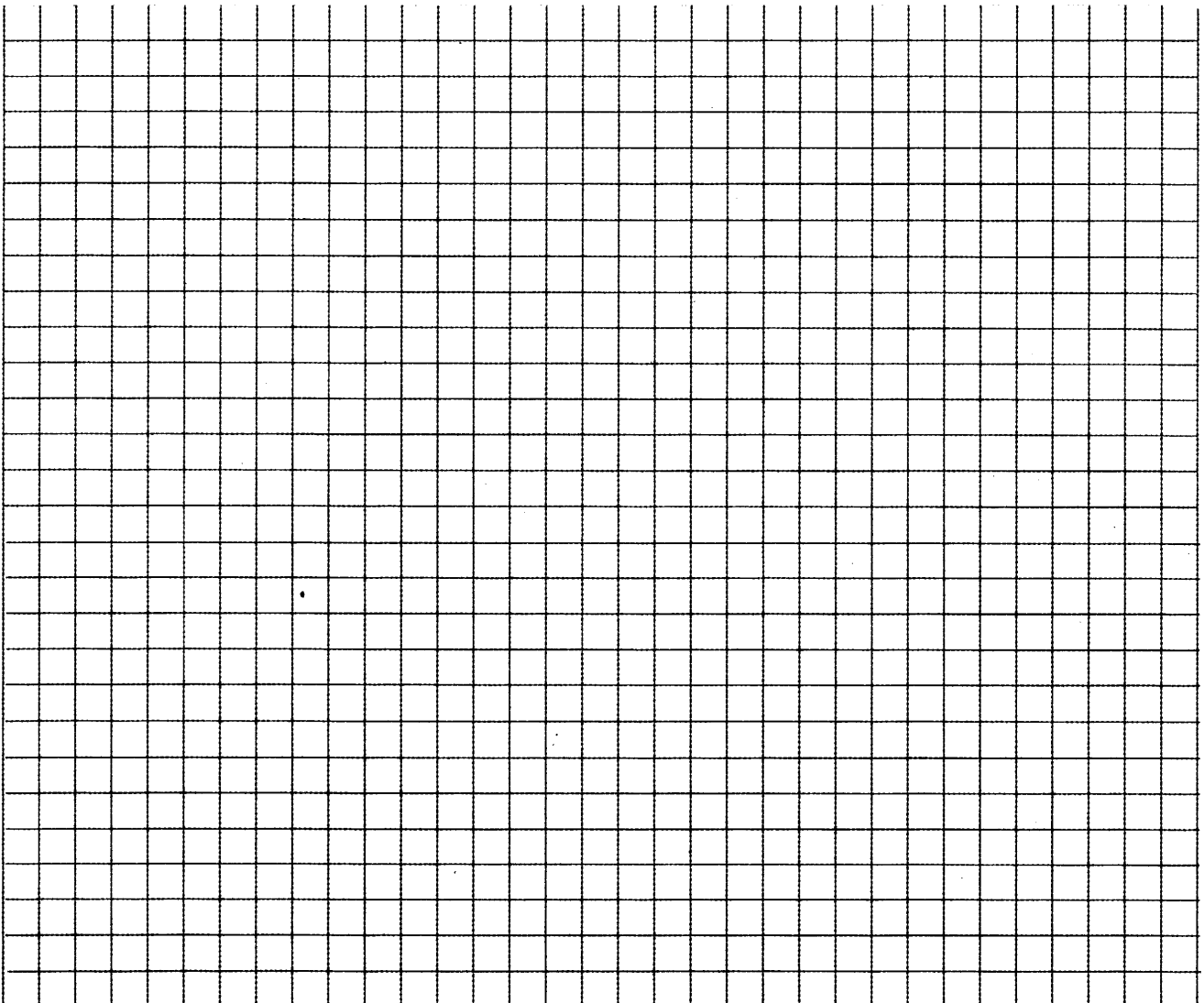


- б) поиск среднего арифметического четных элементов в заданном целочисленном массиве из 50 элементов. Известно, что в массиве есть хотя бы один четный элемент;

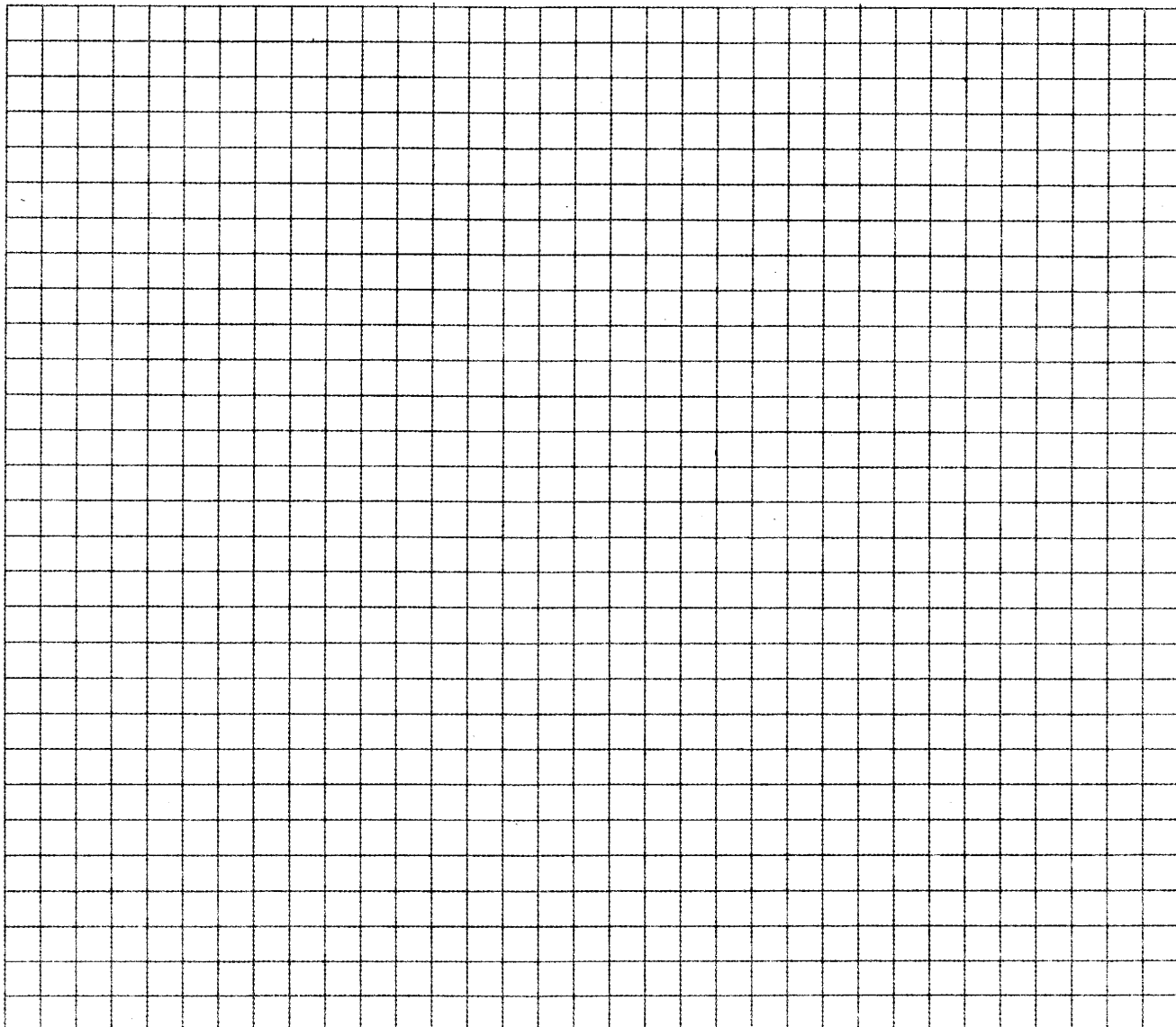




- в)** поиск разности максимально четного и минимально нечетного элементов в заданном целочисленном массиве из 50 элементов. Известно, что в массиве есть и четные, и нечетные элементы;



- г) поиск второго после максимального по величине элемента в заданном целочисленном массиве из 50 различных элементов.



5.23.

- а) Требовалось написать программу, которая решает неравенство « $ax - b < 0$ » относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b); if a = 0 then write('любое число') else if a < 0 then write('x >',b/a) else write('x <', b/a); end.</pre>	<pre>INPUT a, b IF a = 0 THEN PRINT "любое число" ELSE IF a < 0 THEN PRINT "x >",b/a ELSE PRINT "x <",b/a END</pre>	<pre>void main(void) { float a,b,x; scanf("%f%f", &a,&b); if (a==0) printf("любое число"); else if (a<0) printf("x>%f",b/a); else printf("x<%f",b/a); }</pre>

Последовательно выполните два задания:

- 1) приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу;
- 2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

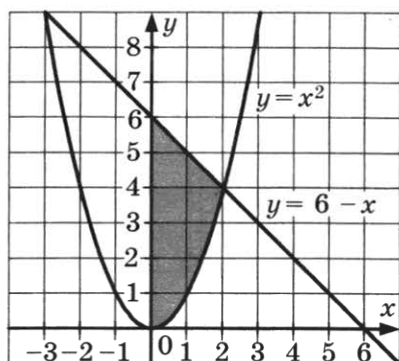
б) Требовалось написать программу, которая решает уравнение « $ax = b$ » относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a, b: real; begin readln(a,b); if b = 0 then write('x = 0') else if a = 0 then write('нет решений') else write('x = ',b/a); end.</pre>	<pre>INPUT a, b IF b = 0 THEN PRINT "x = 0" ELSE IF a = 0 THEN PRINT "нет решений" ELSE PRINT "x=",b/a END</pre>	<pre>void main(void) { float a,b; scanf("%f%f", &a,&b); if (b==0) printf("x=0"); else if (a=0) printf("нет решений"); else printf("x=%f",b/a); }</pre>

Последовательно выполните два задания:

- 1) приведите пример таких чисел a , b , x , при которых программа работает неправильно;
- 2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

в)



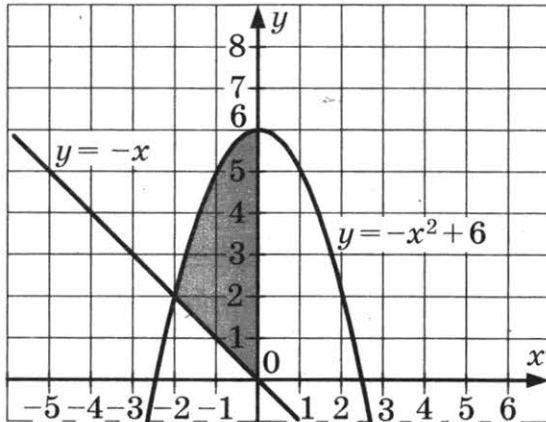
Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x , y — действительные числа) и определяет принадлежность точки закрашенной области, включая ее границы. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x,y: real; begin readln(x,y); if y>=sqrt(x) then if y<=6-x then write('принадлежит') else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF y>=x * x THEN IF y<= 6-x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (y>=x * x) if (y<=6-x) printf("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее:

- 1) приведите пример таких чисел x, y , при которых программа работает неправильно;
- 2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

г)



Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки закрашенной области, включая ее границы. Программист торопился и написал программу неправильно.

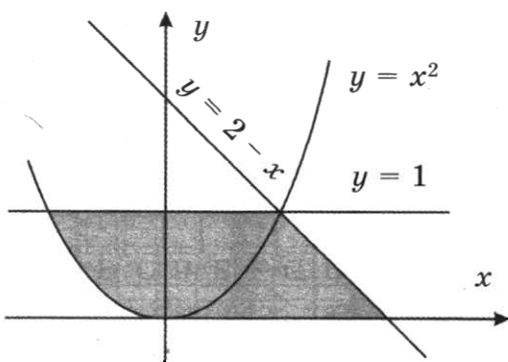
Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x,y: real; begin readln(x,y); if y<=6-sqr(x) then if y>=-x then write('принадлежит') else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF y<=6-x * x THEN IF y>= 6-x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDF ENDF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (y<=6-x * x) if (y>=6-x) printf("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее:

- 1) приведите пример таких чисел x, y , при которых программа работает неправильно;
- 2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

5.24.

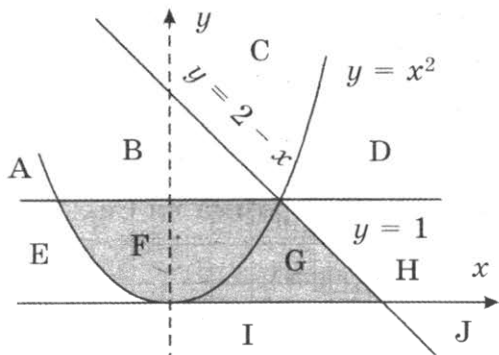
а)



Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки закрашенной области, включая ее границы. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x,y: real; begin readln(x,y); if y>=0 then if y<=1 then if y<=2-x then if y>=x*x then write('принадлежит') else write('не принадлежит') end. end. end. end.</pre>	<pre>INPUT x, y IF y>=0 THEN IF y<= 1 THEN IF y<= 2-x THEN IF y>= x*x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (y>=0) if (y<=1) if (y<=2-x) if (y>=x*x) printf("принадлежит"); else printf("не принадлежит"); } }</pre>

Последовательно выполните следующее:



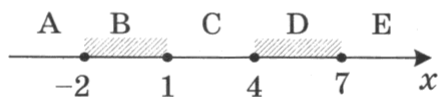
1) перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G, H, I и J). Границы принадлежат заштрихованным областям (F и G).

Область	Условие 1 ($y \geq 0$)	Условие 2 ($y \leq 1$)	Условие 3 ($y \leq 2 - x$)	Условие 4 ($y \geq x^2$)	Программа выведет	Область обрабатывается верно
A						
B						
C						
D						
E						
F						
G						
H						
I						
J						

В столбцах условий укажите «Да», если условие выполнится, «Нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «Не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «Не изв.». В последнем столбце укажите «Да» или «Нет»;

2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

б)



Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (x — действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы). Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x: real; begin readln(x); if x>=-2 then if x<=7 then if x<4 then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x IF x >= -2 THEN IF x <= 7 THEN IF x < 4 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre>	<pre>void main(void) { float x; scanf("%f",&x); if (x>=-2) if (x<=7) if (x<4) printf("принадлежит"); else printf("не принадлежит"); } }</pre>

Последовательно выполните следующее:

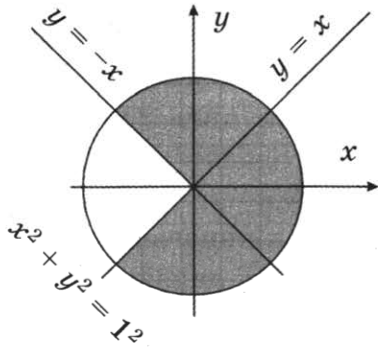
1) перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки — 2, 1, 4 и 7) принадлежат заштрихованным областям (B и D соответственно).

Область	Условие 1 ($x \geq -2$)	Условие 2 ($x \leq 7$)	Условие 3 ($x < 4$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

В столбцах условий укажите «Да», если условие выполнится, «Нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «Не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «Не изв.». В последнем столбце укажите «Да» или «Нет»;

2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

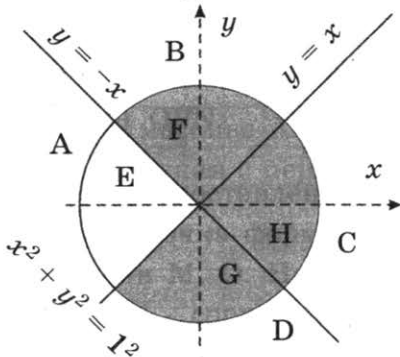
в)



Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки закрашенной области, включая ее границы. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x,y: real; begin readln(x,y); if x*x+y*y<=1 then if y<=x then if y>=-x then write('принадлежит') else write('не принадлежит') end else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF x*x+y*y <= 1 THEN IF y <= x THEN IF y >= -x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (x*x+y*y<=1) if (y<=x) if (y>=-x) printf("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее:



1) перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Границы принадлежат заштрихованным областям (F, G и H).

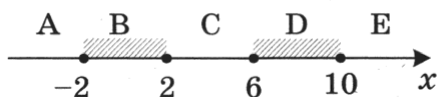
Область	Условие 1 ($x*x+y*y \leq 1$)	Условие 2 ($y \leq x$)	Условие 4 ($y \geq -x$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите «Да», если условие выполнится, «Нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «Не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, при-

надлежащих области, будут выведены разные тексты, напишите «Не изв.». В последнем столбце укажите «Да» или «Нет»;

2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

г)



Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (x — действительное число) и определяется принадлежность этой точки одному из выделенных отрезков В и D (включая границы). Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x: real; begin readln(x); if x<=10 then if x>=-2 then if x>=6 then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x IF x <= 10 THEN IF x >= -2 THEN IF x >= 6 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre>	<pre>void main(void) { float x; scanf("%f",&x); if (x<=10) if (x>=-2) if (x>=6) printf("принадлежит"); else printf("не принадлежит"); } }</pre>

Последовательно выполните следующее:

1) перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D и Е). Границы (точки -2, 2, 6 и 10) принадлежат заштрихованным областям (В и D соответственно).

Область	Условие 1 ($x \leq 10$)	Условие 2 ($x \geq -2$)	Условие 3 ($x \geq 6$)	Программа выведет	Область обрабатывается верно
А					
В					
С					
Д					
Е					

В столбцах условий укажите «Да», если условие выполнится, «Нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «Не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «Не изв.». В последнем столбце укажите «Да» или «Нет»;

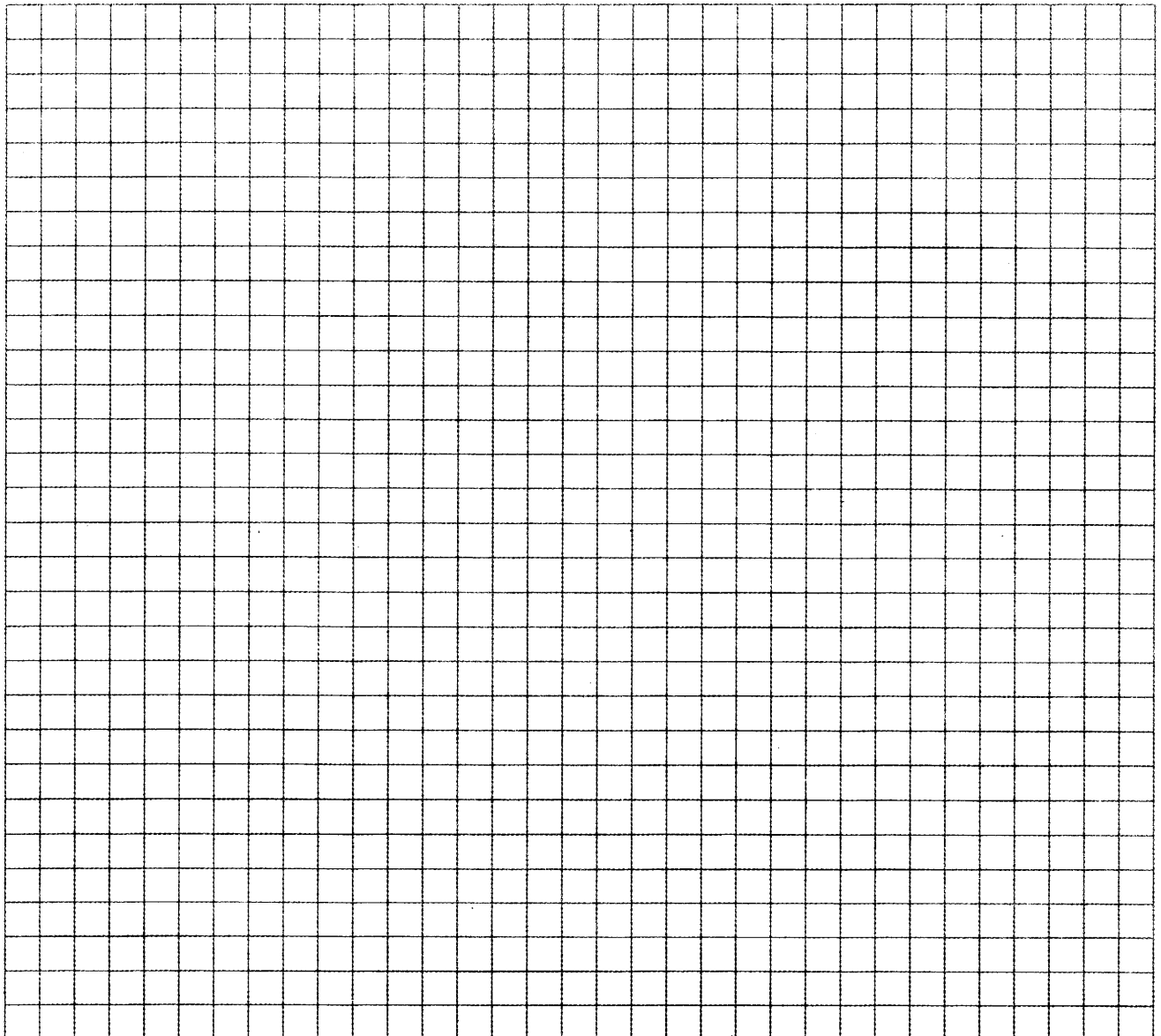
2) укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

5.25.

- а) На вход программе подаются сведения о сотрудниках некоторой организации. В первой строке сообщается количество сотрудников N , которое не меньше 5, но не превосходит 1000, каждая из следующих N строк имеет следующий формат: <Фамилия> <Должность> <Стаж> <Оклад> <Премия>, где
- <Фамилия> — строка, состоящая не более чем из 20 символов;
 - <Должность> — строка, состоящая не более чем из 15 символов;
 - <Стаж> <Оклад> <Премия> — целые числа. Под стажем понимается количество полных лет, отработанных сотрудником в данной организации.
- Оклад и премия — целые числа, не превышающие 1 000 000. Эти элементы входной строки отделены друг от друга одним пробелом. Пример входной строки:

Иванов механик 15 25500 5000

Требуется написать программу (укажите язык программирования, который вы используете), которая будет выводить на экран фамилии, должности и стаж трех сотрудников с наибольшей зарплатой. Если среди остальных есть сотрудники с такой же зарплатой, что и один из этих трех, то следует также вывести и их фамилии, должности и стаж. Зарплата сотрудника равна сумме его оклада и премии.

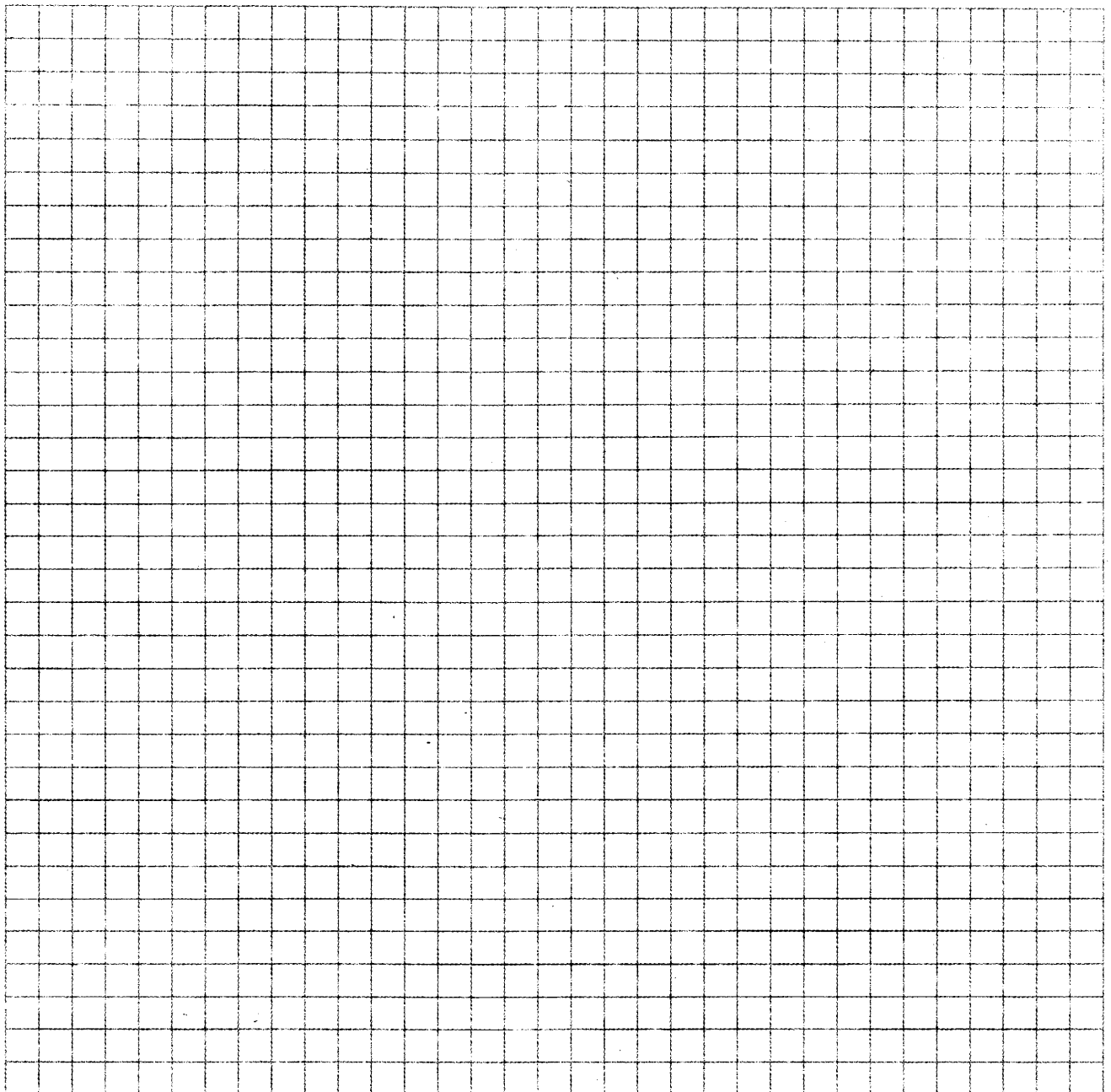


б) На вход программе подаются сведения о сотрудниках некоторой организации. В первой строке сообщается количество сотрудников N , которое не меньше 5, но не превосходит 1000, каждая из следующих N строк имеет следующий формат: <Фамилия> <Должность> <Стаж> <Оклад> <Премия>, где <Фамилия> — строка, состоящая не более чем из 20 символов; <Должность> — строка, состоящая не более чем из 15 символов; <Стаж> <Оклад> <Премия> — целые числа. Под стажем понимается количество полных лет, отработанных сотрудником в данной организации. Оклад и премия — целые числа, не превышающие 1 000 000. Эти элементы входной строки отделены друг от друга одним пробелом. Пример входной строки:

Иванов механик 15 25500 5000

Требуется написать программу (укажите язык программирования, который вы используете), подсчитывающую среднюю зарплату сотрудников, стаж которых больше среднего по организации.

Зарплата сотрудника равна сумме его оклада и премии.



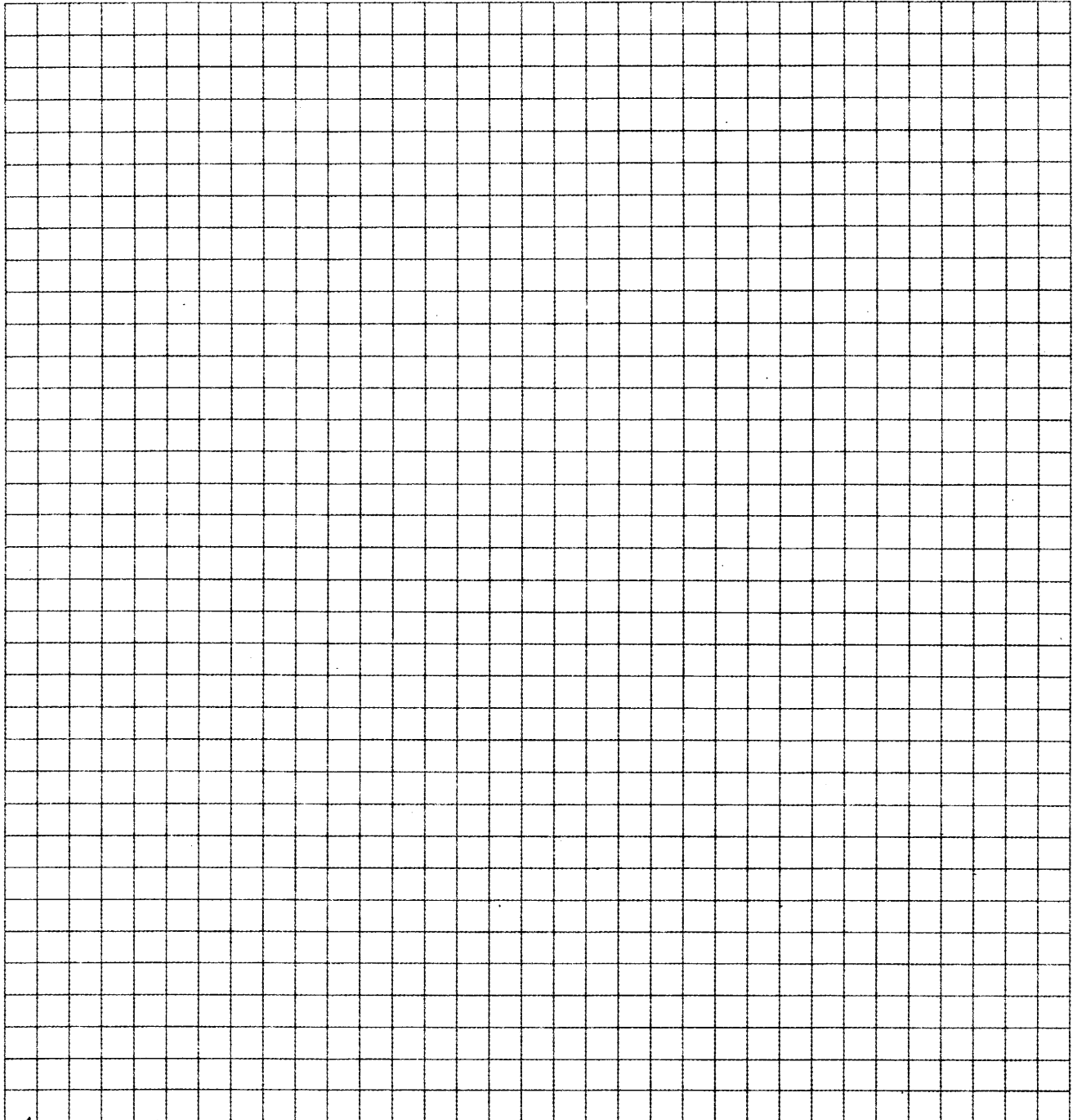
в) На вход программе подаются строчные английские буквы. Ввод этих символов заканчивается точкой (другие символы, отличные от «.» и букв «а» ... «z», во входных данных отсутствуют; в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка).

Требуется написать эффективную программу (укажите язык программирования, который вы используете), которая будет печатать в алфавитном порядке только те буквы, которые ровно три раза подряд встретились во входной последовательности. Каждая буква должна быть распечатана только один раз. Точка при этом не учитывается.

Например, пусть на вход подаются следующие символы: `btfgbbffjrtatbama.`

В данном случае программа должна вывести:

`Aft`



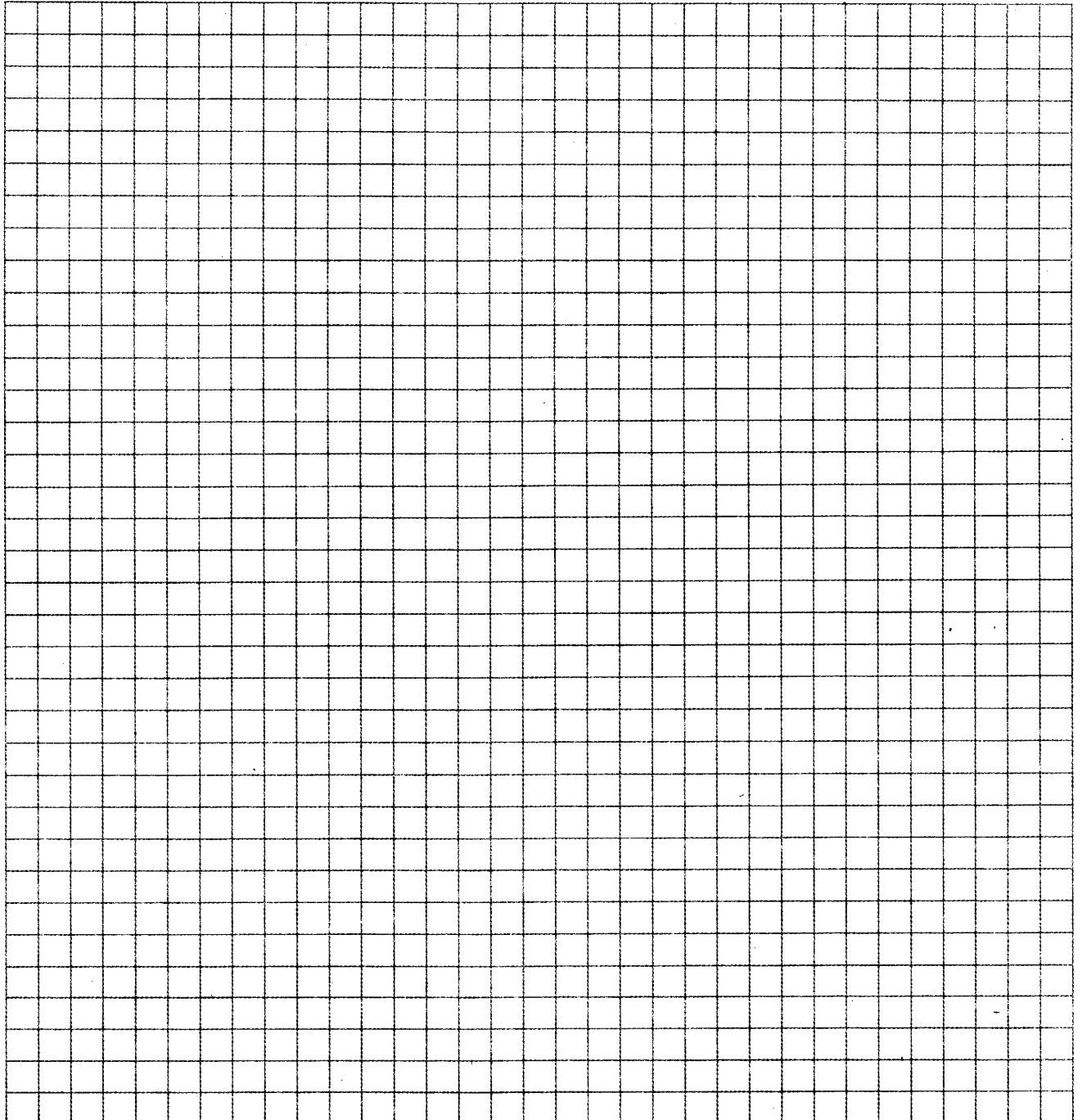
- г) На вход программе подаются строчные английские буквы. Ввод этих символов заканчивается точкой (другие символы, отличные от «.» и букв «a» ... «z», во входных данных отсутствуют; в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка).

Требуется написать эффективную программу (укажите язык программирования, который вы используете), которая будет печатать в алфавитном порядке только те буквы, которые хотя бы однократно ровно три раза подряд встретились во входной последовательности. Каждая буква должна быть распечатана только один раз. Точка при этом не учитывается.

Например, пусть на вход подаются следующие символы: `bbbffjrtatbbbamaaa.`

В данном случае программа должна вывести:

`ab`



5.26. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = F(n-1) + F(n/2), \text{ при } n > 1 \text{ и } n \text{ кратно } 2$$

$$F(n) = F(n-1) + 1, \text{ при } n > 1 \text{ и } n \text{ не кратно } 2$$

$$F(1) = 1$$

Чему равно значение функции $F(8)$?

Рекомендация. Будем вычислять значение функции последовательно, начиная от 1:

$$F(1) = 1 \text{ (дано по условию)}$$

$$F(2) = F(1) + F(1) = 1 + 1 = 2. \text{ (2 кратно 2, по формуле } F(n) = F(n-1) + F(n/2) = F(2-1) + F(2/2).$$

$$F(3) = F(3-1) + 1 = F(2) + 1 = 2 + 1 = 3.$$

$$F(4) = F(4-1) + F(4/2) = F(3) + F(2) = 3 + 2 = 5.$$

5.27. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = F(n-1) + F(n/2), \text{ при } n > 1 \text{ и } n \text{ кратно } 2$$

$$F(n) = F(n-1) + F(n-2), \text{ при } n > 1 \text{ и } n \text{ не кратно } 2$$

$$F(1) = 1$$

Чему равно значение функции $F(8)$?

5.28. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = 2 * F(n/2), \text{ при } n > 2 \text{ и } n \text{ кратно } 2$$

$$F(n) = F(n-1) + 2 * F(n-2), \text{ при } n > 2 \text{ и } n \text{ не кратно } 2$$

$$F(n) = n, \text{ при } n < 3$$

Чему равно значение функции $F(9)$?

5.29. Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N ($N < 10^9$) и выводится на экран максимальная цифра этого числа. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var N,k:integer; begin readln(N); k:=9; while N >= 10 do begin if N mod 10>k then k:=N mod 10; N:=N div 10 end; write(k) end.</pre>	<pre>INPUT N K=9 WHILE N >= 10 IF N MOD 10 > k THEN K=N MOD 10 ENDIF N = N \ 10 WEND PRINT K END</pre>	<pre>void main(void) { int N,k; scanf("%d",&N); k = 9; while (N >= 10) { if (N % 10 > k) k = N % 10; N = N / 10; } printf("%d",k); }</pre>

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 537.
2. Приведите пример такого числа, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки: выпишите строку, которая написана неправильно и приведите правильный вариант строки.

Рекомендация. Для ответа на первый вопрос настоятельно рекомендуем выполнить трассировку предложенного алгоритма для указанного числа. Важно при этом не принимать желаемое за действительное и выполнять именно то, что делает алгоритм, а не то, что он должен бы, вроде бы, в этот момент делать.

Для трассировки нужно составить таблицу, в которой столько же столбцов, сколько переменных описано в программе и, «водя пальцем» по строчкам программы, от начала, последовательно, в порядке выполнения, для каждой строчки менять значение той переменной в ее столбце таблицы, которая этой строчкой программы меняется.

Обычно используют краткую трассировку, для убыстрения. Мы приведем полную таблицу трассировки (в ней мы будем указывать еще и выполняемую в данный момент строку на языке Паскаль), чтобы процесс был наиболее понятен.

<i>N</i>	<i>k</i>	Оператор	Условие
537		readln(N);	
	9	k:=9;	
		while N>=10 do	537>=10, Да
		if N mod 10>k then	7>9, Нет
53		N:=N div 10	
		while N>=10 do	53>=10, Да
		if N mod 10>k then	3>9, Нет
5		N:=N div 10	
		while N>=10 do	5>=10, Нет
		write(k)	На экране: 9

Эта трассировка, кроме ответа на первый вопрос (9), должна нам подсказать, где программа работает неправильно.

Конечно, для решения этой задачи очень желательно уметь самому составлять такой алгоритм. Эта задача относится к одному из вариантов перевода числа в некоторую систему счисления (основанием не больше 10). В данном случае в десятичную систему счисления. А именно: из числа последовательно извлекается младшая цифра (нахождением остатка от деления на 10), после чего эта цифра из числа вычеркивается (целочисленным делением на 10). Потом процесс повторяется, пока в числе не окажется цифр.

Анализируя работу программы, обнаруживаем, что переменная *k* в процессе работы не менялась. Ни одна из цифр числа *N* не оказывается меньше 9. Отсюда находим первую ошибку — начальное значение *k* нужно было сделать не наибольшим из возможных (9, как сейчас), а, наоборот, наименьшим. То есть, нулем.

Получается, что число *k* при любом *N* не будет меняться и программа всегда будет выдавать 9.

Значит, для ответа на второй вопрос (при каком числе программа будет работать верно) достаточно указать любое число, в котором наибольшая цифра и есть 9. Например, просто число 9.

Поиск еще ошибки. Если Вы внимательно выполняли задание 2.16, то, возможно, обратили внимание на то, что процесс деления на 10 нужно продолжать, пока исходное число не станет равным нулю.

А в процессе трассировки Вы, скорее всего, заметили, что старшая (первая) цифра числа *N* (в данном случае 5) вообще не сравнивалась с переменной *k*. Задаемся вопросом: почему? и находим противоречие с тем, как программа должна была работать. Вместо того, чтобы делить число *N* на 10, пока оно не станет равно 0, приведенная программа останавливается, как только число *N* становится меньше 10. Значит, вторая ошибка — неверное условие окончания цикла «while».

Нужно было остановиться, когда число N станет равно нулю. Значит, условие продолжения цикла — пока число N не равно нулю (так как нам известно, что исходное число N натуральное, достаточно написать, что N больше нуля).

Общий план решения задачи:

1. Выполните аккуратно трассировку для предложенного числа.
2. Постарайтесь понять, что в программе помешало получить правильный ответ.
3. Исправьте строку, в которой Вы обнаружили ошибку, и снова выполните трассировку для предложенного числа. Возможно, Вы при этом обнаружите еще одну ошибку.
4. Если в условии имеется пункт, предлагающий подобрать число, при котором программа работает верно, постарайтесь подобрать такое число. Возможно, подойдет такое число, которое удовлетворяет начальному значению искомой величины (если ошибка в начальном значении, как в этом примере). Либо подберите число, при котором цикл выполняется нужное количество раз (если ошибка — в условии окончания цикла). В любом случае еще раз выполните трассировку для выбранного числа и убедитесь, что программа действительно выдает нужный ответ.
5. Аккуратно выпишите те строки, которые содержат ошибку, и приведите правильные варианты этих строк.

5.30. Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N ($N < 10^9$) и выводится на экран минимальная цифра этого числа. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var N,k:integer; begin readln(N); k:=9; while N > 9 do begin if N mod 10 < k then k:=N mod 10; N:=N div 10 end; write(k) end.</pre>	<pre>INPUT N K=9 WHILE N > 9 IF N MOD 10 < k THEN K=N MOD 10 ENDIF N = N \ 10 WEND PRINT K END</pre>	<pre>void main(void) { int N,k; scanf("%d",&N); k = 9; while (N > 9) { if (N % 10 < k) k = N % 10; N = N / 10; } printf("%d",k); }</pre>

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 357.
2. Приведите пример такого числа, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки: выпишите строку, которая написана неправильно и приведите правильный вариант строки.

5.31. Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N ($N < 10^9$) и выводится на экран сумма цифр этого числа. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var N,k:integer; begin readln(N); k:=N mod 10; while N > 1 do begin k:=k+N mod 10; N:=N div 10 end; write(k) end.</pre>	<pre>INPUT N K=N MOD 10 WHILE N > 1 K=K+N MOD 10 N = N \ 10 WEND PRINT K END</pre>	<pre>void main(void) { int N,k; scanf("%d",&N); k = N % 10; while (N > 1) { k = k + N % 10; N = N / 10; } printf("%d",k); }</pre>

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 357.
2. Приведите пример такого числа, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки: выпишите строку, которая написана неправильно, и приведите правильный вариант строки.

- 5.32.** Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N ($N < 10^9$) и выводится на экран произведение цифр этого числа. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var N,k:integer; begin readln(N); k:=0; while N > 0 do begin k:=k*N mod 10; N:=N div 10 end; write(k) end.</pre>	<pre>INPUT N K = 0 WHILE N > 0 K=K*N MOD 10 N = N \ 10 WEND PRINT K END</pre>	<pre>void main(void) { int N,k; scanf("%d",&N); k = 0; while (N > 0) { k = k * N % 10; N = N / 10; } printf("%d",k); }</pre>

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 257.
2. Приведите пример такого числа, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки: выпишите строку, которая написана неправильно и приведите правильный вариант строки.

- 5.33.** У исполнителя Удвоитель две команды, которым присвоены номера:

1. Прибавь 1,
2. Умножь на 2.

Первая из них увеличивает число на экране на 1, вторая увеличивает число на экране вдвое. Программа для Удвоителя — это последовательность команд. Сколько есть программ, которые число 1 преобразуют в число 21?

Решение.

Будем последовательно вычислять количество программ ($F(N)$), которые число 2 преобразуют в число N . Для этого будем заполнять таблицу:

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
$F(N)$																					

Число программ, которые преобразуют число 1 в число 1, равно 1. Это пустая программа (не содержит ни одной инструкции). Запишем это значение в ячейку для $F(1)$. Теперь будем последовательно, начиная со следующей ячейки ($N = 2$), анализировать, из каких чисел может быть получено число N . Так, например, $N = 2$ может быть получено из $N = 1$ прибавлением 1, или из $N = 1$ умножением на 2. Получаем, что $F(2) = F(1) + F(1) = 1 + 1 = 2$. Запишем в ячейку $F(2)$.

При данных командах (+1 и *2) очевидно, что нечетные N могут быть получены только командой «+1» из предыдущего N (из $N - 1$). Значит, для них мы будем записывать то же число, что и в соседней слева ячейке. То есть, $F(3) = F(2) = 2$. А четные N могут быть получены также путем умножения $N/2$ на 2. То есть, $F(N) = F(N - 1) + F(N/2)$. Например, $F(4) = F(4 - 1) + F(4/2) = F(3) + F(2) = 2 + 2 = 4$. Таким образом, последовательно заполняя ячейки, получим:

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
$F(N)$	1	2	2	4	4	6	6	10	10	14	14	20	20	26							

5.34. У исполнителя Удвоитель две команды, которым присвоены номера:

1. Прибавь 1,
2. Умножь на 2.

Первая из них увеличивает число на экране на 1, вторая увеличивает число на экране вдвое. Программа для Удвоителя — это последовательность команд. Сколько есть программ, которые число 2 преобразуют в число 25?

5.35. У исполнителя Удвоитель две команды, которым присвоены номера:

1. Прибавь 1,
2. Умножь на 2.

Первая из них увеличивает число на экране на 1, вторая увеличивает число на экране вдвое. Программа для Удвоителя — это последовательность команд. Сколько есть программ, которые число 4 преобразуют в число 29?

5.36. Ниже на четырех языках программирования записан рекурсивный алгоритм F .

Бейсик	Паскаль
<pre>SUB F(n) PRINT n IF n <= 4 THEN F(n * 2) F(n + 1) END IF END SUB</pre>	<pre>procedure F(n: integer); begin writeln(n); if n <= 4 then begin F(n * 2); F(n + 1); end end</pre>

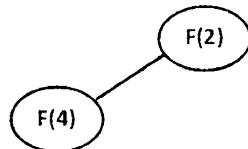
Си	Алгоритмический язык
<pre>void F(int n) { printf("%d\n", n); if (n <= 4) { F(n * 2); F(n + 1); } }</pre>	<pre>алг F(цел n) нач вывод n, нс если n <= 4 то F(n * 2) F(n + 1) все кон</pre>

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова $F(2)$?

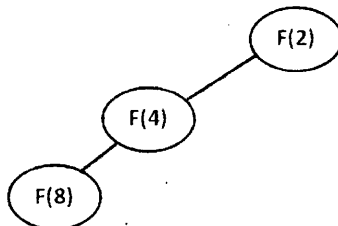
Решение.

Выполнение рекурсивного алгоритма можно представить следующим образом: каждый рекурсивный вызов процедуры F порождает в памяти компьютера новую копию этой процедуры и запускает ее на выполнение со своими значениями входных параметров. После того как процедура F завершила работу (возможно, в процессе работы ей пришлось породить и вызвать несколько своих копий с соответствующими входными параметрами), выполнение программы продолжается со следующего оператора после вызова F . Здесь важно понимать, что разные копии процедуры F , порожденные одним рекурсивным вызовом, могут одновременно находиться в памяти компьютера на разных этапах своего выполнения и иметь различные значения своих локальных переменных. Для наглядности представим схему рекурсивных вызовов в виде дерева. Поскольку в процедуре F имеются два рекурсивных вызова $F(n*2)$ и $F(n+1)$, то наше дерево будет двоичным. Каждый узел дерева будет соответствовать вызову процедуры $F(x)$. Потомками каждого узла дерева будут вызовы из процедуры F , соответствующей узлу. Левым потомком будет вызов вида $F(n*2)$, правым — $F(n+1)$. Корнем дерева будет самый первый вызов $F(2)$.

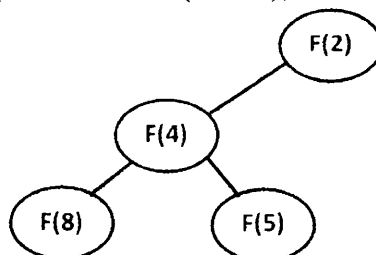
Вызвали $F(2*2) = F(4)$, сразу напечатали 2.



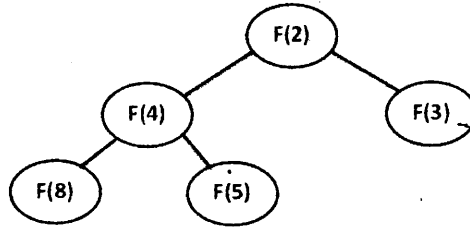
Напечатали 4. Затем проверяем условие ($4 \leq 4$ — истина), поэтому пошел следующий вызов $F(4*4) = F(8)$



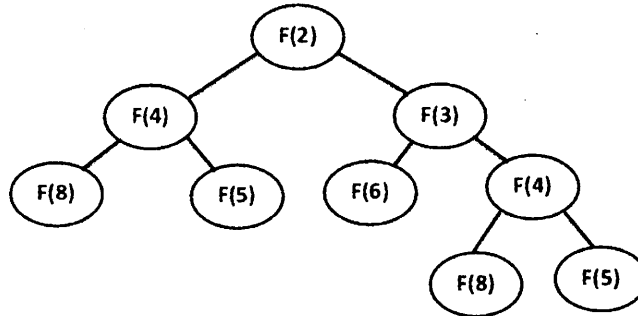
Напечатали 8. Проверяем условие ($8 \leq 4$ — ложь), поэтому тело условного оператора не выполняется, данная копия процедуры F заканчивает работу и выполнение программы продолжается в копии $F(4)$ с оператора вызова $F(4+1)$, т.е. с $F(5)$.



Напечатали 5. Проверяем условие ($5 \leq 4$ – ложь), поэтому тело условного оператора не выполняется, данная копия процедуры F заканчивает работу и выполнение программы продолжается в копии F(2) с оператора вызова F(2 + 1), т.е. с F(3).



Действуя аналогично, получаем полное дерево всех рекурсивных вызовов



Нетрудно заметить, что поскольку значение входного параметра процедуры печатается каждый раз при вызове процедуры, искомая сумма равна сумме чисел в узлах дерева

$$8 + 4 + 5 + 2 + 3 + 6 + 4 + 8 + 5 = 45.$$

Замечание 1. Построенное дерево позволяет ответить на более сложный вопрос: «что напечатает программа?» Выписав значения узлов в порядке построения, получим: 2 4 8 5 3 6 4 8 5

Замечание 2. Результат работы программы при ином расположении оператора печати n , в общем случае, отличается от данного. Например, такая процедура F (Паскаль)

```

procedure F(n: integer);
begin
  if n <= 4 then
    begin
      writeln(n);
      F(n * 2);
      F(n + 1);
    end
  end
end
  
```

напечатает только значения, большие или равные четырем (при том же дереве рекурсивных вызовов), т.е. 2 4 3 4, и сумма их будет, соответственно, 13. Будьте внимательны при выполнении подобных заданий.

Второй способ решения. Так как нас интересует только итоговая сумма, будем считать, что рекуррентная функция результатом своей работы выдает на экран определенную сумму чисел. Например, будем считать, что $F(n)$ — сумма чисел, которые будут выведены на экран

при вызове $F(n)$. Будем последовательно вычислять значение этой функции (аналогично задаче 5.33).

Для начала составим рекуррентные выражения по приведенной программе.

Легко заметить, что если $n > 4$, то $F(n) = n$ (подпрограмма выводит только число n и больше ничего не делает, потому что условие $n \leq 4$ не выполняется).

А если $n \leq 4$, то $F(n) = n + F(2n) + F(n + 1)$. Потому что подпрограмма выводит на экран число n , а также вызывает $F(2*n)$ и $F(n + 1)$, которые, в свою очередь, добавляют на экран соответствующую сумму чисел. Все эти числа мы и складываем.

Применяем полученные формулы, начиная с тех значений $F(n)$, которые очевидны и встречаются при выполнении программы. Так как нам нужно вычислить $F(2)$, которая зависит от функции для больших значений n , будем вычислять $F(n)$ от больших значений к меньшим. Начнем с $F(8)$, так как наибольшее значение n , при котором будет вызываться $F(n)$ — это $F(4*2)$.

$$F(8) = 8$$

$$F(7) = 7$$

$$F(6) = 6$$

$$F(5) = 5$$

Для всех этих $F(n)$ результат был очевиден, потому что $n > 4$. Следующие значения $F(n)$ будем вычислять по формуле для $n \leq 4$: $F(n) = n + F(2n) + F(n + 1)$.

$$F(4) = 4 + F(2*4) + F(4 + 1) = 4 + F(8) + F(5) = 4 + 8 + 5 = 17$$

$$F(3) = 3 + F(2*3) + F(3 + 1) = 3 + F(6) + F(4) = 3 + 6 + 17 = 26$$

$$F(2) = 2 + F(2*2) + F(2 + 1) = 2 + F(4) + F(3) = 2 + 17 + 26 = 45$$

Получили тот же ответ — 45. При наличии на экзамене свободного времени рекомендуем решать задачу обоими способами, чтобы можно было сверить ответы.

5.37. Ниже на четырех языках программирования записан рекурсивный алгоритм F .

Бейсик	Паскаль
<pre>SUB F(n) IF n <= 4 THEN F(n * 2) F(n + 1) END IF PRINT n END SUB</pre>	<pre>procedure F(n: integer); begin if n <= 4 then begin F(n * 2); F(n + 1); end; writeln(n); end</pre>
Си	Алгоритмический язык
<pre>void F(int n) { if (n <= 4) { F(n * 2); F(n + 1); } printf("%d\n", n); }</pre>	<pre>алг F(цел n) нач если n <= 4 то F(n * 2) F(n + 1) все ВЫВОД n, нс кон</pre>

Решение.

Анализируя принцип работы Чертежника, замечаем, что его перемещение по каждой из координат (x, y) не зависит от перемещения по другой координате. Значит, перемещение по каждой координате можно рассматривать независимо.

Запишем суммарное перемещение Чертежника по оси X после выполнения всей программы: $3 + n*(a + 27) - 27$. Так как цикл выполняется n раз, команды $+a$ и $+27$ учитываем n раз (умножаем $(a + 27)$ на n). Так как известно, что Чертежник вернулся в ту же точку, это суммарное перемещение равно нулю. Получаем: $3 + n*(a + 27) - 27 = 0$.

Аналогично по оси Y получаем: $3 + n*(b + 12) - 12 = 0$.

В обоих уравнениях оставим n , умноженное на что-то, в левой части, а остальное перенесем в правую часть и вычислим.

Получим:

$$n*(a + 27) = 24$$

$$n*(b + 12) = 9$$

Так как n — количество повторений цикла, n — целое неотрицательное число.

Из первого уравнения получаем, что n является делителем числа 24, из второго — делителем числа 9. У этих двух чисел (24 и 9) общими делителями являются только числа 1 и 3. Числа 1 среди предлагаемых ответов нет. Значит, это число 3.

5.42. Исполнитель Чертежник перемещается на координатной плоскости, оставляя след в виде линии. Чертежник может выполнять команду **сместиться на (a, b)** , где a, b — целые числа. Эта команда перемещает Чертежника из точки с координатами (x, y) в точку с координатами $(x + a; y + b)$.

Например, если Чертежник находится в точке с координатами $(4, 2)$, то команда **сместиться на $(2, -3)$** переместит Чертежника в точку $(6, -1)$.

Цикл

ПОВТОРИ число РАЗ

последовательность команд

КОНЕЦ ПОВТОРИ

означает, что последовательность команд будет выполнена указанное число раз (число должно быть натуральным).

Чертежнику был дан для исполнения следующий алгоритм (буквами n, a, b обозначены неизвестные числа):

НАЧАЛО

сместиться на $(3, 5)$

ПОВТОРИ n РАЗ

сместиться на (a, b)

сместиться на $(15, 5)$

КОНЕЦ ПОВТОРИ

сместиться на $(-17, -40)$

КОНЕЦ

После выполнения этого алгоритма Чертежник возвратился в исходную точку. Какое из приведенных ниже чисел обозначено буквой n ?

1) 7

2) 2

3) 3

4) 4

5.43. Ниже на четырех языках приведена программа. Определить, при скольких различных значениях k программа печатает тот же результат, что и при $k = 100$ (включая $k = 100$).

Паскаль	Си
<pre>function f(n:integer):integer; begin f := n * n * n end; var k,i :integer; BEGIN readln(k); i:=12; while f(i)>k do i:=i-1; write(i); END.</pre>	<pre>int f(int n) { return n * n * n; } void main() { int k, i; scanf("%d",&k); i = 12; while (f(i)>k) i--; printf("%d", i); }</pre>
Бейсик	Алгоритмический язык
<pre>DIM K, I AS INTEGER INPUT K I = 12 WHILE F(I) > K I = I - 1 WEND PRINT I FUNCTION F(N) F = N * N * N END FUNCTION</pre>	<pre>алг нач цел k, i ввод k i := 12 нц пока F(i) > k i := i - 1 кц вывод i кон алг цел F(цел n) нач знач := n * n * n кон</pre>

Решение. Сначала вычислим, какой результат выдает программа при указанном $k = 100$. Функция $f(n)$ вычисляет куб числа n . Цикл WHILE уменьшает значение переменной i от 12 и меньше, пока $f(i) > k$. То есть, пока $i^3 > 100$. Как только это перестает выполняться, цикл оканчивается. То есть, как только i^3 становится ≤ 100 . Самое большое значение i , при котором $i^3 \leq 100$ — это 4 ($4^3 = 64$, 5^3 уже $125 > 100$). Чтобы найти все такие k , при которых программа выводит тот же ответ, нужно найти все такие k , при которых $4^3 \leq k$, а $5^3 > k$. Получаем неравенство: $64 \leq k < 125$. Нам нужны все целые значения k . Это от 64 до 124. То есть, $124 - 64 + 1 = 61$.

5.44. Ниже на четырех языках приведена программа. Определить, при скольких различных значениях k программа печатает тот же результат, что и при $k = 150$ (включая $k = 150$).

Паскаль	Си
<pre>function f(n:integer):integer; begin f := n * n * n * n end; var k,i :integer; BEGIN readln(k); i:=12; while f(i)>k do i:=i-1; write(i); END.</pre>	<pre>int f(int n) { return n * n * n * n; } void main() { int k, i; scanf("%d",&k); i = 12; while (f(i)>k) i--; printf("%d", i); }</pre>

Бейсик	Алгоритмический язык
<pre> DIM K, I AS INTEGER INPUT K I = 12 WHILE F(I) > K I = I - 1 WEND PRINT I FUNCTION F(N) F = N * N * N * N END FUNCTION </pre>	<pre> алг нач цел k, i <u>ВВОД</u> k i := 12 <u>нц пока</u> F(i) > k i := i - 1 <u>кц</u> <u>ВЫВОД</u> i. кон алг <u>цел</u> F(цел n) нач <u>знач</u> := n * n * n * n кон </pre>

5.45. Получив на вход число x , алгоритм печатает два числа L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 12, а потом 3.

Паскаль	Си
<pre> var x, L, M: integer; begin readln(x); L := 0; M := 0; while x > 0 do begin M := M + 1; L := L + x mod 10; x := x div 10; end; writeln(L); write(M); end. </pre>	<pre> #include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L = 0; M = 0; while (x > 0) { M = M + 1; L = L + x % 10; x = x / 10; } printf("%d\n%d", L, M); } </pre>
Бейсик	Алгоритмический язык
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X > 0 M = M + 1 L = L + X MOD 10 X = X \ 10 WEND PRINT L PRINT M </pre>	<pre> алг нач цел x, L, M <u>ВВОД</u> x L := 0 M := 0 <u>нц пока</u> x > 0 M := M + 1 L := L + mod(x, 10) x := div(x, 10) <u>кц</u> <u>ВЫВОД</u> L, <u>нс</u>, M кон </pre>

Решение. Анализируем, что делает алгоритм. Исходное число x в цикле, пока оно больше нуля, делят на 10 и что-то делают с остатками от деления на 10. Это известный алгоритм перевода числа в другую систему счисления (в данном случае — в десятичную). То есть, вы-

деление десятичных цифр из числа. На каждом шаге цикла переменная M увеличивается на 1. Это счетчик количества итераций цикла. Сказано, что он становится равен 3-м. Значит, было выделено 3 цифры числа. Значит, число x трехзначное. Во время работы цикла цифры числа добавляются к переменной L . Это сумма цифр числа. Сказано, что она становится равна 12. Остается только подобрать такое наибольшее трехзначное число, сумма цифр которого равна 12-ти. Чтобы число было наибольшим, первой цифрой поставим самое большое, что можно, — это цифра 9. Сумма оставшихся цифр остается равной 3-м ($12 - 9 = 3$). Вторая цифра тоже должна быть наибольшей из возможных. Это цифра 3. Последняя цифра остается — 0. Ответ — 930.

5.46. Получив на вход число x , алгоритм печатает два числа L и M . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 3, а потом 16.

Паскаль	Си
<pre>var x, L, M: integer; begin readln(x); L := 9; M := 0; while x>0 do begin M := M + x mod 10; if x mod 10 < L then L := x mod 10; x := x div 10; end; writeln(L); write(M); end.</pre>	<pre>#include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L = 9; M = 0; while (x > 0) { M = M + x % 10; if (x % 10 < L) L = x % 10; x = x / 10; } printf("%d\n%d", L, M); }</pre>
Бейсик	Алгоритмический язык
<pre>DIM X, L, M AS INTEGER INPUT X L = 9 M = 0 WHILE X > 0 M = M + X MOD 10 IF X MOD 10 < L THEN L = X MOD 10 END IF X = X \ 10 WEND PRINT L PRINT M</pre>	<pre>алг нач цел x, L, M ВВОД x L := 9 M := 0 нц пока x > 0 M := M + mod(x,10) если mod(x,10) < L то L := mod(x,10) все x := div(x,10) кц ВЫВОД L, M кон</pre>

5.47. На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число R следующим образом.

- 1) Строится двоичная запись числа N .
- 2) К этой записи дописываются справа еще два разряда по следующему правилу:

- а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
- б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите минимальное число R , которое превышает 43 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Решение. Наименьшее число, которое превышает 43, — число 44. Переведем его в двоичную систему счисления. Получим 101100_2 . Проверим, может ли оно быть получено приведенным алгоритмом. В результате алгоритма к двоичному представлению числа добавляется две двоичные цифры. Возьмем первые цифры числа, кроме последних двух и применим к ним алгоритм. В результате действия а) получаем $1 + 0 + 1 + 1 = 3$. В остатке от деления на 2 — 1. А в двоичной записи числа 44 на этом месте стоит цифра 0. Значит, число 44 не подходит. Однако то, что оно не подходит, еще не означает, что начальная часть числа не подходит. Первую цифру мы уже вычислили. Вычислим вторую: $1 + 0 + 1 + 1 + 1 = 4$. Остаток от деления на 2 равен 0. Получаем двоичное число 101110_2 . Переведем его в десятичную систему. Получаем 46. Так как мы получили его по приведенному алгоритму и оно больше 43, это и есть ответ.

5.48. То же условие, что и в 5.47, но нужно найти минимальное число, превышающее 41.

Решение. Переведем число 42 в двоичную систему. Получаем 101010_2 . Применяем алгоритм к первым цифрам 1010_2 . Получаем 101000_2 . Это число 40. Оно нам не подходит, так как меньше 41. Наименьшее число, которое может нас устроить — следующее. То есть, оно получается из числа, следующего после 1010_2 . Это число 1011_2 . Применим к нему алгоритм. Получим 101110_2 . Это 46.

5.49. На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число V следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа еще два разряда по следующему правилу:

- а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
- б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа V .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 46. В ответе это число запишите в двоичной системе счисления.

Рекомендации. Аналогично предыдущей задаче, берем наименьшее число, большее 46 (это 47) и переводим его в двоичную систему счисления (получится, кстати, 6 разрядов). Далее проверяем, верно ли, что по первым 4-м разрядам можно посредством алгоритма получить оставшиеся два разряда. Если совпало — эти 4 разряда и есть ответ. Если не совпало — сравниваем полученное двухразрядное двоичное число с тем, что стоит в конце числа 47. Если полученные 2 разряда больше имеющихся в числе 47 — первые 4 разряда и есть ответ. Если больше — в ответ записываем следующее двоичное 4-разрядное число (можно получить прибавлением 1).

5.50. На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1) Строится двоичная запись числа N .

2) К этой записи дописываются справа еще два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите минимальное число R , которое превышает 56 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

5.51. На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число V следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа еще два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа V .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 49. В ответе это число запишите в двоичной системе счисления.

5.52. На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число V следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа еще два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа V .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 54. В ответе это число запишите в двоичной системе счисления.

5.53. Напишите в ответе число, равное количеству различных значений входной переменной k , при которых приведенная ниже программа выводит тот же ответ, что и при входном значении $k = 10$. Значение $k = 10$ также включается в подсчет различных значений k . Для Вашего удобства программа приведена на пяти языках программирования.

Бейсик	Питон
<pre> DIM K, I AS LONG INPUT K I = 1 WHILE F(I) < K I = I + 1 WEND IF F(I)-K <= K-F(I-1) THEN PRINT I ELSE PRINT I-1 END IF FUNCTION F(N) F = N * N * N END FUNCTION </pre>	<pre> def f(n): return n*n*n i = 1 k = int(input()) while f(i) < k: i+=1 if (f(i)-k <= k-f(i-1)): print (i) else: print (i - 1) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел i, k ввод k i := 1 нц пока f(i) < k i := i + 1 кц если f(i)-k <= k-f(i-1) то вывод i иначе вывод i-1 все кон алг цел f(цел n) нач знач := n * n * n кон </pre>	<pre> var k, i : longint; function f(n: longint) : longint; begin f := n * n * n; end; begin readln(k); i := 1; while f(i) < k do i := i + 1; if f(i)-k <= k-f(i-1) then writeln(i) else writeln(i-1); end. </pre>
Си	
<pre> #include <stdio.h> long f(long n) { return n * n * n; } void main() { long k, i; scanf("%ld", &k); i = 1; while (f(i) < k) i++; if (f(i)-k <= k-f(i-1)) printf("%ld", i); else printf("%ld", i-1); } </pre>	

Рекомендации.

Сначала вычислим, что выводит программа при $k = 10$.

В цикле переменная i меняется от 1 на 1, пока $i^3 < 10$. То есть, при некотором i выражение $i^3 < 10$ перестанет выполняться (то есть, будет $i^3 \geq 10$). $2^3 = 8$, $3^3 = 27$. Значит, это при $i = 3$.

Подставим это $i = 3$ в следующее условие: $f(i) - k \leq k - f(i - 1)$. $3^3 - 10 \leq 10 - 2^3$

$27 - 10 \leq 10 - 8$? Нет.

Значит, на экран будет выведено $(i - 1)$. То есть, 2.

Теперь найдем, при каких k программа будет выводить на экран 2.

Это может быть при выводе $(i - 1)$, как в рассмотренном случае при $k = 10$. А также при выводе i (если условие $f(i) - k \leq k - f(i - 1)$ выполнится).

Рассмотрим первый случай. На экран выводится $(i - 1)$. Условие не выполнилось.

При этом $i = 3$ и условие будет: $3^3 - k > k - 2^3$. $27 - k > k - 8$. $35 > 2k$. $k < 17,5$. Так как k — целое число, то это условие равносильно $k \leq 17$.

Это мы рассмотрели условие вывода на экран. Еще нужно, чтобы после окончания цикла переменная i была равна 3. То есть, при $i = 2$ условие цикла выполняется: $f(i) < k$. $2^3 < k$. $k > 8$.

А при $i = 3$ условие цикла перестает выполняться: $f(i) \geq k$. $3^3 \geq k$. $k \leq 27$.

Объединяем (вернее, пересекаем) все три условия ($k \leq 17$, $k > 8$, $k \leq 27$) и получаем $8 < k \leq 17$.

Рассмотрим второй случай. На экран выводится (i) . Условие выполнилось.

При этом $i = 2$ и условие будет: $2^3 - k \leq k - 1^3$. $8 - k \leq k - 1$. $9 \leq 2k$. $k \geq 4,5$. Так как k — целое число, то это условие равносильно условию $k \geq 5$.

Чтобы после окончания цикла переменная i была бы равна 2, нужно, чтобы при $i = 1$ условие цикла еще выполнялось: $1^3 < k$. $k > 1$.

А при $i = 2$ условие цикла должно перестать выполняться: $2^3 \geq k$. $k \leq 8$.

Объединяем (вернее, пересекаем) все три условия ($k \geq 5$, $k > 1$, $k \leq 8$) и получаем $5 \leq k \leq 8$.

Объединяем условия из первого и второго случаев и получаем $5 \leq k \leq 17$. От 5 до 17 получаем 13 различных значений переменной k .

5.54. Напишите в ответе число, равное количеству различных значений входной переменной k , при которых приведенная ниже программа выводит тот же ответ, что и при входном значении $k = 30$. Значение $k = 30$ также включается в подсчет различных значений k . Для Вашего удобства программа приведена на пяти языках программирования.

Бейсик	Питон
<pre>DIM K, I AS LONG INPUT K I = 1 WHILE F(I) < K I = I + 1 WEND IF F(I) - F(I-1) > K THEN PRINT I ELSE PRINT I-1 END IF FUNCTION F(N) F = N * N - 30 END FUNCTION</pre>	<pre>def f(n): return n*n-30 i = 1 k = int(input()) while f(i) < k: i+=1 if (f(i) - f(i-1) > k): print (i) else: print (i - 1)</pre>

Алгоритмический язык	Паскаль
<pre> алг нач цел i, k ввод k i := 1 нц пока f(i) < k i := i + 1 кц если f(i) - f(i-1) > k то вывод i иначе вывод i-1 все кон алг цел f(цел n) нач знач := n * n - 30 кон </pre>	<pre> var k, i : longint; function f(n: longint) : longint; begin f := n * n - 30; end; begin readln(k); i := 1; while f(i) < k do i := i + 1; if f(i) - f(i-1) > k then writeln(i) else writeln(i-1); end. </pre>

Си

```

#include <stdio.h>
long f(long n)
{
  return n * n - 30;
}
void main()
{
  long k, i;
  scanf("%ld", &k);
  i = 1;
  while (f(i) < k)
    i++;
  if (f(i) - f(i-1) > k)
    printf("%ld", i);
  else
    printf("%ld", i-1);
}

```

5.55. Напишите в ответе число, равное количеству различных значений входной переменной k , при которых приведенная ниже программа выводит тот же ответ, что и при входном значении $k = 40$. Значение $k = 40$ также включается в подсчет различных значений k . Для Вашего удобства программа приведена на пяти языках программирования.

Бейсик	Питон
<pre> DIM K, I AS LONG INPUT K I = 100 WHILE F(I) > K I = I - 1 WEND IF F(I) - F(I-1) > K - 20 THEN PRINT I ELSE PRINT I-1 END IF FUNCTION F(N) F = N * (N - 1) END FUNCTION </pre>	<pre> def f(n): return n*(n-1) i = 100 k = int(input()) while f(i) > k: i-=1 if (f(i) - f(i-1) > k - 20): print (i) else: print (i - 1) </pre>

Алгоритмический язык	Паскаль
<pre> алг нач цел i, k ввод k i := 100 нц пока f(i) > k i := i - 1 кц если f(i) - f(i-1) > k - 20 то вывод i иначе вывод i-1 все кон алг цел f(цел n) нач знач := n * (n - 1) кон </pre>	<pre> var k, i : longint; function f(n: longint) : longint; begin f := n * (n - 1); end; begin readln(k); i := 100; while f(i) > k do i := i - 1; if f(i) - f(i-1) > k - 20 then writeln(i) else writeln(i-1); end. </pre>
Си	
<pre> #include <stdio.h> long f(long n) { return n * (n - 1); } void main() { long k, i; scanf("%ld", &k); i = 100; while (f(i) > k) i++; if (f(i) - f(i-1) > k - 20) printf("%ld", i); else printf("%ld", i-1); } </pre>	

5.56. Исполнитель Апрель15 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1,
2. Умножить на 2.

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Апрель15 — это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 21 и при этом траектория вычислений содержит число 10?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Решение. Так как по условию «траектория вычислений» должна содержать число 10, получаем, что число 10 должно обязательно содержаться в последовательности преобразований числа 1 в число 21. То есть нужно отдельно посчитать количество вариантов получения числа 10 из числа 1 и количество вариантов получения числа 21 из числа 10, а после перемножить результаты. И то и другое можно сделать уже знакомым нам способом. Например, по-

строим таблицу, в которой будем последовательно вычислять число вариантов получения числа n из числа 1 (где n будет меняться от 1 до 10). Обозначим $F(n)$ — число вариантов получения числа n из числа 1. Очевидно, что $F(1) = 1$ (число различных программ, которые состоят из приведенных команд 1 и 2 и получают из числа 1 число 1 — одна (в ней нет ни одной команды)). Если n — нечетное, то $F(n) = F(n - 1)$.

А если n — четное, то $F(n) = F(n - 1) + F(n/2)$.

Строим таблицу:

n	1	2	3	4	5	6	7	8	9	10
$F(n)$	1	2	2	4	4	6	6	10	10	14

Число вариантов получения из 10 числа 21 проще посчитать, построив дерево возможных преобразований. Оно получится весьма примитивным, потому что после первого же действия ($10 + 1 = 11$ и $10 \cdot 2 = 20$) получаются числа, из которых 21 можно получить только одним способом — прибавляя единицу. То есть, из 10 можно получить 21 только двумя различными программами.

Перемножаем ответы $14 \cdot 2 = 28$.

Заметим, что так как при построении таблицы возможны арифметические ошибки, рекомендуем для проверки построить также дерево возможных преобразований из 1 в 10 и убедиться, что число вариантов в дереве совпадает с числом вариантов, полученных таблицей.

5.57. Исполнитель Июль20 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1,
2. Умножить на 2.

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Июль20 — это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 24 и при этом траектория вычислений содержит число 8?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

5.58. Исполнитель Июль20 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1,
2. Умножить на 2.

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Июль20 — это последовательность команд.

Сколько существует программ, для которых при исходном числе 2 результатом является число 37 и при этом траектория вычислений содержит число 15?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

5.59. Исполнитель Июль20 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1,
2. Умножить на 2.

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Июль20 — это последовательность команд.

Сколько существует программ, для которых при исходном числе 3 результатом является число 39 и при этом траектория вычислений содержит число 17?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

5.60. Исполнитель Редактор получает на вход строку цифр и преобразовывает ее. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) заменить (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды

заменить (222, 34)

преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v , то выполнение команды **заменить (v, w)** не меняет эту строку.

Б) нашлось (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

ТО команда1

ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 95 идущих подряд цифр 7? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (777) ИЛИ нашлось (888)

ЕСЛИ нашлось (777)

ТО заменить (777, 8)

ИНАЧЕ заменить (888, 7)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Решение. Выполним приведенный алгоритм.

На первом этапе каждые три первых 777 меняются на одну 8.

В строке исходно 31 тройка 777. Поэтому в результате получится строка, в которой в начале 31 цифра 8, затем оставшиеся две цифры 7.

Теперь алгоритм будет заменять первые 888 на 7. Затем вторые 888 на 7. Затем третьи 888 на 7.

После этого в строке снова найдется 777, которые будут заменены на 8.

То есть из 31-й восьмерки сначала 9 первых будет заменено на 777. Затем эти 777 снова на 8.

То есть получим в строке сначала $31 - 9 + 1 = 23$ восьмерки, затем 77.

На следующем этапе, аналогично, в строке останется на 8 восьмерок меньше (15 восьмерок и 77).

На следующем этапе — еще на 8 меньше. Получится 7 восьмерок и 77.

Теперь первые 888 поменяются на 7 и вторые 888 поменяются на 7.

Получится 77877. В этой строке нет ни одной группы 777 или 888. Поэтому алгоритм закончится.

Ответ — 77877.

5.61. Исполнитель Редактор получает на вход строку цифр и преобразовывает ее. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) заменить (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды

заменить (222, 34)

преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v , то выполнение команды **заменить (v, w)** не меняет эту строку.

Б) нашлось (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

ТО команда1

ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

а) Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 75 идущих подряд цифр 7? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (777) ИЛИ нашлось (888)

ЕСЛИ нашлось (777)

ТО заменить (777, 8)

ИНАЧЕ заменить (888, 7)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

б) Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 69 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (777) ИЛИ нашлось (888)

ЕСЛИ нашлось (777)

ТО заменить (777, 8)

ИНАЧЕ заменить (888, 7)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

6. ИНФОРМАЦИОННЫЕ МОДЕЛИ

Основные понятия

В задачах ЕГЭ на анализ информационных моделей обычно используются две основные *информационные модели* — таблицы и схемы. Как правило, умения понимать эти модели достаточно для решения данного рода задач.

Информация в *таблице* построена по следующему принципу: на пересечении строки и столбца находится информация, характеризующая комбинацию этой строки и столбца.

Пример. Таблица умножения. На пересечении строки номер X и столбца Y находится результат умножения X на Y .

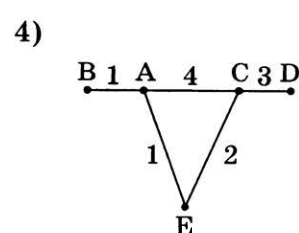
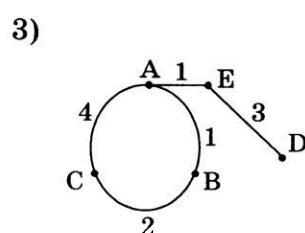
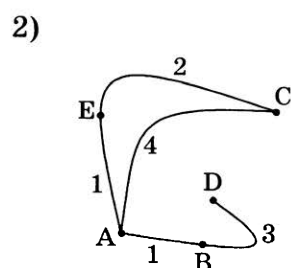
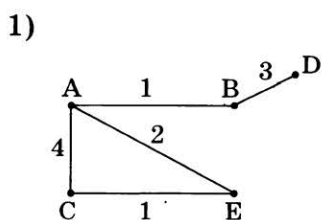
Информация на схеме построена по следующему принципу: если между объектами схемы имеется связь, она отображается линией, соединяющей названия этих объектов на схеме.

Пример. Схема железнодорожных станций (или метрополитена). Между двумя станциями можно проехать, если на схеме нарисована линия, соединяющая эти станции.

Практическая часть

6.1. В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

	A	B	C	D	E
A		1	4		1
B	1			3	
C	4				2
D		3			
E	1		2		



Рекомендации. Анализируем представленную информацию. Таблица отображает стоимость для всех комбинаций станций. То есть на пересечении строки X и столбца Y находится число, соответствующее стоимости проезда от станции X до станции Y . Отсутствие числа в ячейке соответствует отсутствию дороги между этими станциями.

На схеме названия станций соединены линиями с числами. Судя по всему, линия, соединяющая станцию X со станцией Y , отражает наличие пути между станциями X и Y , а число на этой линии — стоимость проезда по этому пути.

Задача нахождения соответствия между схемой и таблицей, таким образом, сводится к задаче идентичности связей и их стоимостей.

Так, по таблице из станции A можно добраться до станции B за 1, до станции C за 4, и до станции E за 1.

Схема 1 не соответствует этому (путь из А в Е «стоит» 2). Дальше не будем ее рассматривать. Схемы 2, 3 и 4 соответствуют.

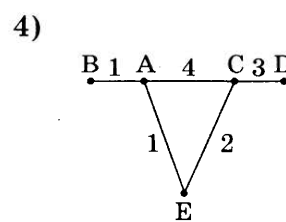
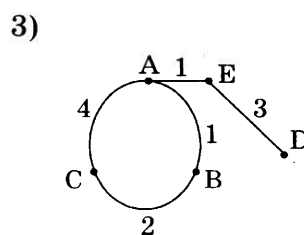
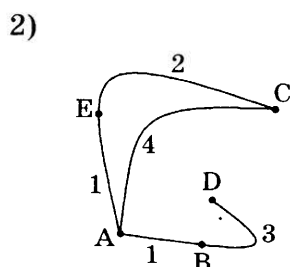
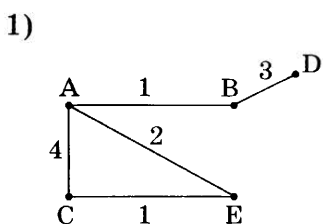
По таблице от станции В можно добраться до станции В за 1, а до станции D за 3.

Схема 2 соответствует этому. Схемы 3 и 4 не соответствуют этому (путь из В в D на них отсутствует). Значит, единственным возможным кандидатом остается только схема 2.

Для подстраховки можно проверить на соответствие остальные строки таблицы. Если обнаружится несоответствие, значит, либо ошиблись составители задачи (что очень маловероятно, ибо все задания по несколько раз проверяются), либо мы были невнимательны в одной из наших предыдущих проверок и отсеяли что-то правильное.

6.2. В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

	A	B	C	D	E
A		1	4		1
B	1		2		
C	4	2			
D					3
E	1			3	



6.3. Таблица стоимости перевозок устроена следующим образом: числа, стоящие на пересечениях строк и столбцов таблиц, означают стоимость проезда между соответствующими соседними станциями. Если пересечение строки и столбца пусто, то станции не являются соседними.

Укажите таблицу, для которой выполняется условие: «Минимальная стоимость проезда из А в В не больше 6».

Стоимость проезда по маршруту складывается из стоимостей проезда между соответствующими соседними станциями.

1)

	A	B	C	D	E
A			3	1	
B			4	2	
C	3	4			2
D	1				
E		2	2		

2)

	A	B	C	D	E
A			3	1	1
B			4		
C	3	4			2
D	1				
E	1		2		

3)

	A	B	C	D	E
A			3	1	
B			4		1
C	3	4			2
D	1				
E		1	2		

4)

	A	B	C	D	E
A				1	
B			4		1
C		4		4	2
D	1		4		
E		1	2		

Рекомендации. Если таблица большая и плотно заполнена числами, то для решения такого рода задач требуется, вообще говоря, сложный алгоритм (например, алгоритм Дейкстры). Его изучение не входит в стандартный курс школьной информатики. Но в данном случае таблица небольшая и далеко не все ее клетки заполнены. Значит, можно ее достаточно эффективно решить, осуществив осмысленный перебор возможных вариантов.

При решении рекомендуется воспользоваться двумя соображениями:

- 1) вариант пути, который становится больше или равен 6, но еще не достигает конечной точки (В), можно дальше не рассматривать;
- 2) возвращение в точку, через которую уже прошел путь (петля), не приведет к эффективному решению, поэтому такие ветки тоже можно не рассматривать.

Рассмотрим пример такого «осмысленного перебора»:

Таблица 1:

Из точки А два пути — в точку С (равен 3) и в точку D (равен 1).

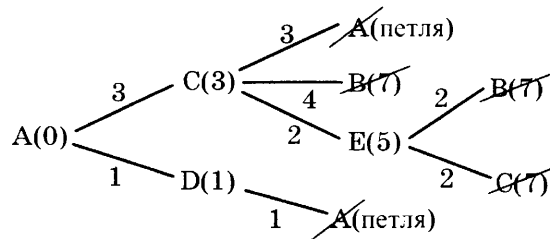
Если пошли в точку С (путь А—С), из нее три пути — в точку А (петля, не рассматриваем), в точку В (равен 4, в сумме $3 + 4 = 7$, больше 6, не рассматриваем) и в точку Е (равен 2, итого путь А — С—Е равен $3 + 2 = 5$).

Из точки Е только два пути, оба равны 2, то есть сумма $5+2$ будет больше 6, дальше можно не рассматривать.

Если пошли в точку D (путь А—D), из нее только путь в точку А (петля).

Все возможные пути мы отбросили, до точки В за 6 добраться не удалось, значит, таблица варианта 1 нам не подходит.

Все это решение гораздо удобнее и эффективнее рисовать при помощи небольшой схемки, на которой рисуются возможные пути из стартовой точки и для каждой точки, в которую приходит путь, указывается текущая стоимость проезда:



Далее нужно таким же образом рассмотреть остальные 3 варианта таблицы. Один из них должен соответствовать условию.

Возможно решение этой задачи в обратную сторону, «с конца». В этом случае сначала рассматриваются все точки, из которых можно добраться до конечной точки (В). Потом — все точки, из которых можно добраться до этих, и т.д. Смысл решения точно такой же, нужно отбрасывать варианты с «петлями» и варианты, стоимость проезда в которых становится больше или равна 6.

Эти два способа решения («с начала» и «с конца») в обычном случае одинаковы. Мы рекомендуем предпочесть решение «с конца», если количество путей, входящих в конечную точку, меньше, чем количество путей, выходящих из стартовой (как, например, в таблице варианта 2).

6.4. Грунтовая дорога проходит последовательно через населенные пункты А, В, С и D. При этом длина дороги между А и В равна 80 км, между В и С — 50 км и между С и D — 10 км.

Между А и С построили новое асфальтовое шоссе длиной 40 км. Оцените минимально возможное время движения велосипедиста (в часах) из пункта А в пункт В, если его скорость по грунтовой дороге — 20 км/час, по шоссе — 40 км/час?

Ответ:	
--------	--

Рекомендация. Метод решения этой задачи такой же, как у предыдущих.

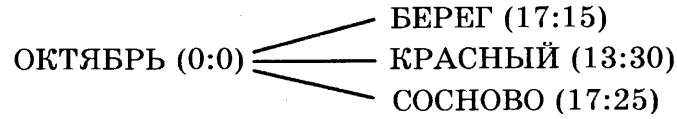
Предлагается рассматривать все возможные варианты, отбрасывая при этом те, которые хуже, чем известные.

Из стартовой точки (ОКТЯБРЬ) можно улететь в БЕРЕГ (прилет в 17:15), КРАСНЫЙ (прилет в 13:30) и в СОСНОВО (прилет в 17:25).

Значит, на данный момент нам известен маршрут, доставляющий путешественника в пункт назначения (СОСНОВО) в 17:25. Это значит, что при рассмотрении остальных маршрутов мы будем отбрасывать те рейсы, которые заканчиваются позже, чем в 17:25.

Кроме того, не имеет смысла рассматривать рейсы, которые отправляются раньше, чем путешественник оказывается в этом аэропорту. Также не имеет смысла возвращаться в ту точку, где путешественник уже был.

Изобразим полученное на схеме:



Из пункта БЕРЕГ есть рейсы в СОСНОВО и в ОКТЯБРЬ. Рейс в СОСНОВО отбрасываем, так как он отправляется в 12:15, а путешественник окажется в БЕРЕГЕ только в 17:15. Рейс в ОКТЯБРЬ не рассматриваем («петля»). Значит, из ОКТЯБРЯ в БЕРЕГ лететь не нужно.

Из КРАСНОГО есть рейсы в ОКТЯБРЬ и в СОСНОВО. Первый не рассматриваем — «петля», второй отправляется раньше, чем в 13:30 (время прибытия путешественника в КРАСНЫЙ).

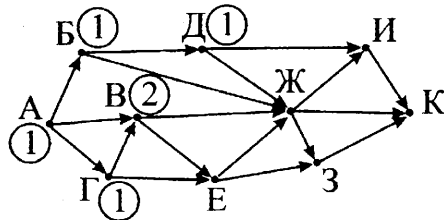
Получаем, что в данной задаче самый лучший вариант — лететь прямым рейсом из ОКТЯБРЯ в СОСНОВО.

6.7. Между четырьмя местными аэропортами: ВОСТОРГ, ЗАРЯ, ОЗЕРНЫЙ и ГОРКА — ежедневно выполняются авиарейсы. Приведен фрагмент расписания перелетов между ними. Путешественник оказался в аэропорту ВОСТОРГ в полночь (0:00). Определите самое раннее время, когда он может попасть в аэропорт ГОРКА.

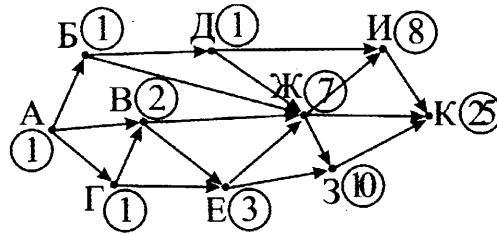
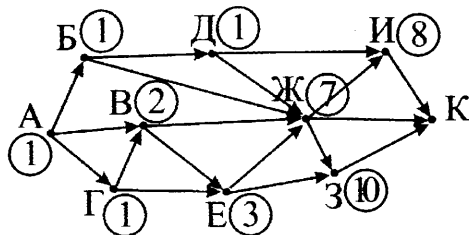
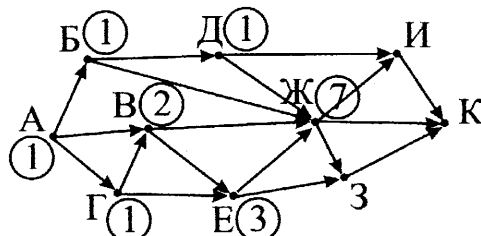
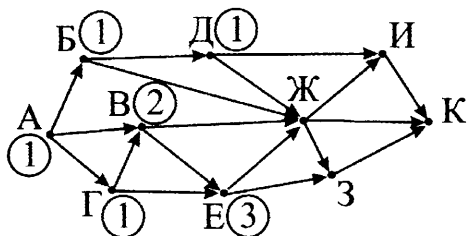
Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
ВОСТОРГ	ГОРКА	16:15	18:30
ОЗЕРНЫЙ	ЗАРЯ	13:40	15:50
ОЗЕРНЫЙ	ВОСТОРГ	14:10	16:20
ГОРКА	ОЗЕРНЫЙ	17:05	19:20
ВОСТОРГ	ОЗЕРНЫЙ	11:15	13:20
ЗАРЯ	ОЗЕРНЫЙ	16:20	18:25
ВОСТОРГ	ЗАРЯ	14:00	16:15
ЗАРЯ	ГОРКА	16:05	18:15
ГОРКА	ЗАРЯ	14:10	16:25
ОЗЕРНЫЙ	ГОРКА	18:35	19:50

Ответ:

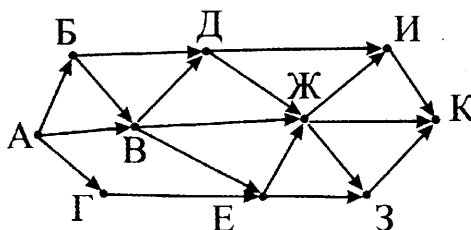
Теперь повторим эту операцию. Найдем все вершины, для которых известны числа для всех входящих «стрелок». Теперь это В и Д. Для вершины Д стрелка опять одна, значит, число путей из вершины А совпадает с предыдущей вершиной (Б), то есть 1. А вот в вершину В входит 2 «стрелки». Поэтому нужно сложить числа всех вершин, из которых идут «стрелки» в данную вершину. То есть $1 + 1 = 2$.



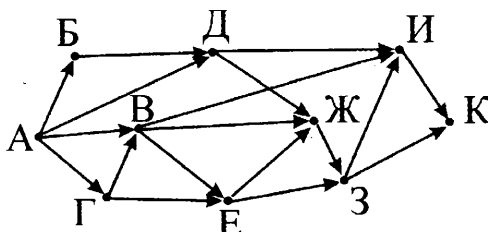
Будем повторять эту операцию для новых вершин. Теперь это можно сделать для вершины Е. В нее входит две «стрелки». Сумма чисел для предыдущих вершин равна $2 + 1 = 3$. Так постепенно получим результат для нужной нам вершины К.



6.12. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



6.13. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



7. ОПРЕДЕЛЕНИЕ ВЫИГРЫШНОЙ СТРАТЕГИИ ИГРЫ (АНАЛИЗ И ПОСТРОЕНИЕ ДЕРЕВА ИГРЫ)

Основные понятия

В третьей части экзамена выделяется одна задача, не связанная с программированием. На данный момент имеется два различных вида этих заданий. В первом случае предлагаются правила некоторой игры для двоих игроков. У каждого игрока имеется обычно 3–4 варианта хода. Нужно выяснить, кто выигрывает при безошибочной игре и как для этого он должен ходить.

При решении заданий данного типа предполагается, что вы проанализируете возможные ходы обоих игроков и определите, кто из них выигрывает и как он для этого должен ходить.

Сначала рассмотрим *пример* решения задачи (самый трудный из тех, что были предложены к настоящему моменту).

7.1. Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 4, а во второй — 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то кучке, или добавляет 2 камня в какую-то кучку. Игра завершается в тот момент, когда количество камней в одной из кучек становится не менее 19. Если в момент завершения игры общее число камней в двух кучках не менее 35, то выиграл Ваня, в противном случае — Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Самым универсальным способом решения задачи является, как ни странно, прямое решение — нужно последовательно, «уровень за уровнем» строить дерево игры. На первый взгляд эта идея выглядит угрожающе сложной (при 4-х вариантах хода, с каждым новым уровнем число рассматриваемых вариантов увеличивается в 4 раза; на 5-м ходу нужно рассматривать уже $4^5 = 1024$ варианта). Не нужно этого заранее бояться. Нужно понимать, что задача составлена так, чтобы она имела разумное решение. А чтобы дерево не «росло» слишком быстро, нужно некоторые ветки отбрасывать, не рассматривая дальше. Это нужно делать в случае, когда дальнейшие ходы, очевидно, приводят к выигрышу одного из игроков.

Рассмотрим это решение. Для удобства построения дерева рекомендуем взять вертикально ориентированный лист бумаги и начать строить дерево не сверху вниз, как это принято, а слева направо. Эта рекомендация вызвана тем, что запись текущей позиции игры занимает много места по горизонтали, а рассматривать придется действительно много вариантов.

Договоримся, что текущее положение игры будем записывать двумя числами, через запятую — сначала число камней в первой кучке, потом — во второй. Так, исходную позицию в данном примере запишем как 4,3. Так как каждый игрок может для обеих кучек утроить число камней или добавить 2 камня, рассмотрим все 4 возможных хода первого игрока:

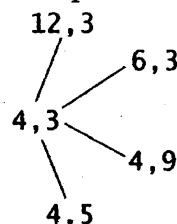


Рисунок 7.1. Дерево игры. Варианты первого хода

Позиция 12,3 получилась после утроения числа камней в первой кучке, позиция 6,3 — после добавления в первую кучку двух камней, позиция 4,9 — после утроения второй кучки, позиция 4,5 — после добавления во вторую кучку двух камней.

Теперь рассмотрим варианты хода второго игрока для каждой полученной позиции:

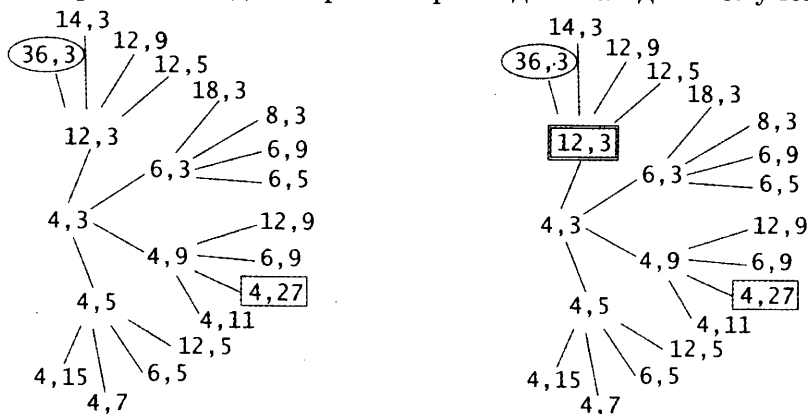


Рисунок 7.2. Дерево игры. Варианты второго хода

Позиция 4,27 обведена прямоугольником, потому что в этой позиции игра заканчивается (во второй кучке больше 19 камней) проигрышем Вани (сумма камней в кучках менее 35). Поэтому Ваня в эту позицию ходить не будет.

Позиция 36,3 обведена эллипсом, потому что в этой позиции игра заканчивается выигрышем Вани. Это значит, что если Петя пойдет в 12,3, то Ваня выиграет ходом 36,3. Значит, позиция 12,3 является для Пети проигрышной и он туда ходить не будет. Обозначим эту позицию двойным прямоугольником. Очевидно, можно не рассматривать дальше ходы из позиции 12,3.

Интересно, что если бы мы заметили это раньше, можно было бы не рассматривать другие ходы из позиции 12,3.

Постараемся проанализировать, как еще можно сократить дерево ходов. Какие ходы нужно рассмотреть в первую очередь? Вероятно, те, про которые сразу будет что-то известно. Те, куда игрок обязательно пойдет, чтобы выиграть, и те, куда игрок ни в коем случае не пойдет, потому что проигрывает.

Сейчас третий ход игры (ход Пети). Если сделать в одной кучке хотя бы 19 камней, игра закончится. Сначала посмотрим, получится ли при этом позиция, при которой сумма камней в кучках будет менее 35 камней (выигрыш Пети). Просматриваем варианты сверху вниз, например, см. рисунок 7.3.

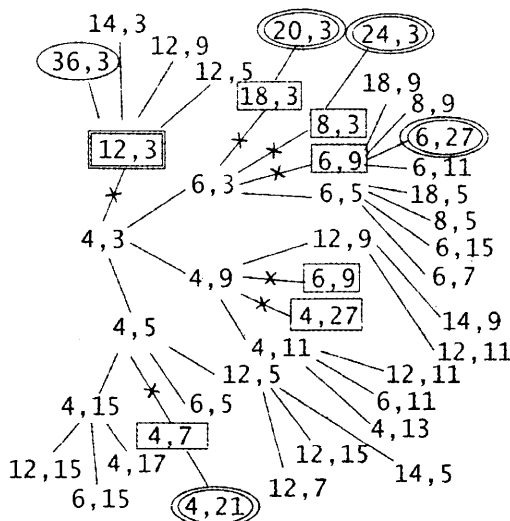


Рисунок 7.3. Дерево игры. Варианты третьего хода

Напоминаем, позиции после 12,3 (14,3 и 12,9 и 12,5) мы более не рассматриваем, потому что если Петя первым ходом пойдет в 12,3, Ваня пойдет обязательно в 36,3 и выиграет.

Следующая позиция — 18,3. Из нее можно закончить игру добавив 2 камня к 18-ти. При этом получится 20,3, это сразу приведет к выигрышу Пети. Значит, позиция 18,3 является выигрышной для Пети. Значит, Ване туда желательно не ходить. Помечаем ее прямоугольником. Очевидно, что остальные 3 варианта хода Пети из 18,3 можно не рассматривать (если есть один выигрышный ход, незачем рассматривать остальные).

Следующая позиция — 8,3. Если утроить 8 камней, получится 24,3. Это снова выигрыш Пети. Значит, и в позицию 8,3 Ване лучше не ходить. Помечаем ее прямоугольником.

Из позиции 6,9 игра закончится только ходом в 6,27. Сумма камней 33. Выигрыш Пети. Значит, в позицию 6,9 Ване тоже лучше не ходить. Помечаем ее прямоугольником.

Из позиции 6,5 нет ни одного способа закончить игру. Рассматриваем все 4 варианта.

Из позиции 12,9 утроение любой кучки приводит к проигрышу Пети. Поэтому рассматриваем только добавление к кучкам двух камней.

Позицию 6,9 второй раз не рассматриваем — она такая же, как уже рассмотренная. Помечаем ее прямоугольником.

Из позиции 4,11 утроение второй кучки приводит к проигрышу Пети. Рассматриваем только остальные 3 варианта хода.

Из позиции 12,5 — аналогично, не рассматриваем 36,5.

Позицию 6,5 второй раз не рассматриваем — она такая же, как уже рассмотренная.

Из позиции 4,7 у Пети есть выигрышный ход в 4,21. Значит, в позицию 4,7 Ване лучше не ходить. Помечаем ее прямоугольником.

Из позиции 4,15 не рассматриваем проигрышный для Пети ход в 4,45. Рассматриваем три оставшиеся хода.

Получившиеся позиции оцениваем с точки зрения хода Вани. Найдем позиции, из которых Ваня одним ходом выигрывает. Заметим, что 18,9 и 8,9 и 6,11 далее не рассматриваем, потому что Петя вместо них пойдет в 6,27 и выиграет.

Рассматриваем 4 варианта хода Пети из 6,5.

Из 18,5 и 6,15 Ваня утроит большую кучку и выиграет. Петя туда ходить не будет. Помечаем их двойным прямоугольником.

Ходы из 8,5 и из 6,7 рассмотрим все (два из них — 24,5 и 6,21 — сразу приводят к выигрышу Пети, далее их не будем рассматривать).

Рассмотрим варианты ходов Вани на ходы Пети из 12,9 и 4,11 и 12,5.

Очевидно, что Ваня может утроить большую кучку в любой из этих позиций и выиграть. Значит, все эти позиции являются для Пети проигрышными. Значит, Ваня из 4,5 пойдет именно в 12,5 и остальные его ходы из 4,5 можно не рассматривать.

Получаем, что позиции 4,5 и 4,9 также являются для Пети проигрышными (см. рис. 7.4).

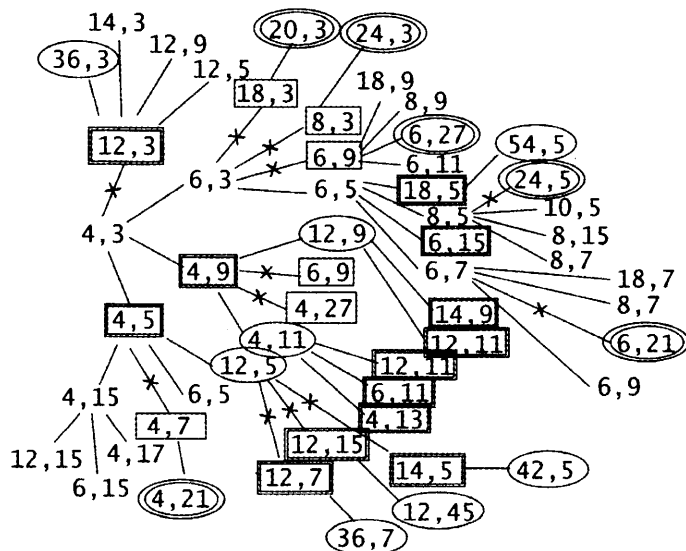


Рисунок 7.4. Дерево игры. Варианты четвертого хода

Осталось рассмотреть только 6 вариантов хода, полученных из позиций 6,7 и 8,5.

Из позиции 18,7 Петя может добавить 2 камня в первую кучку и выиграть (20,7).

Из позиции 8,7 Петя может утроить первую кучку и выиграть (24,7).

Позицию 6,9 мы уже рассматривали: Петя может утроить вторую кучку и выиграть (6,27).

Получается, что при любом ходе Вани из позиции 6,7 у Пети есть ответный выигрышный ход. Значит, позиция 6,7 является для Вани проигрышной. Отсюда делаем вывод, что предыдущая Петина позиция (6,5) — выигрышная. Отсюда делаем вывод, что предыдущая позиция Вани (6,3) — проигрышная для него. Значит, из стартовой позиции Петя должен пойти именно в 6,3, чтобы выиграть.

Варианты хода Пети, полученных из позиции 8,5, теперь можно не рассматривать — мы уже нашли победителя.

Мы обозначили толстыми линиями те ходы, которые должен делать Петя, чтобы выиграть (см. рис. 7.5).

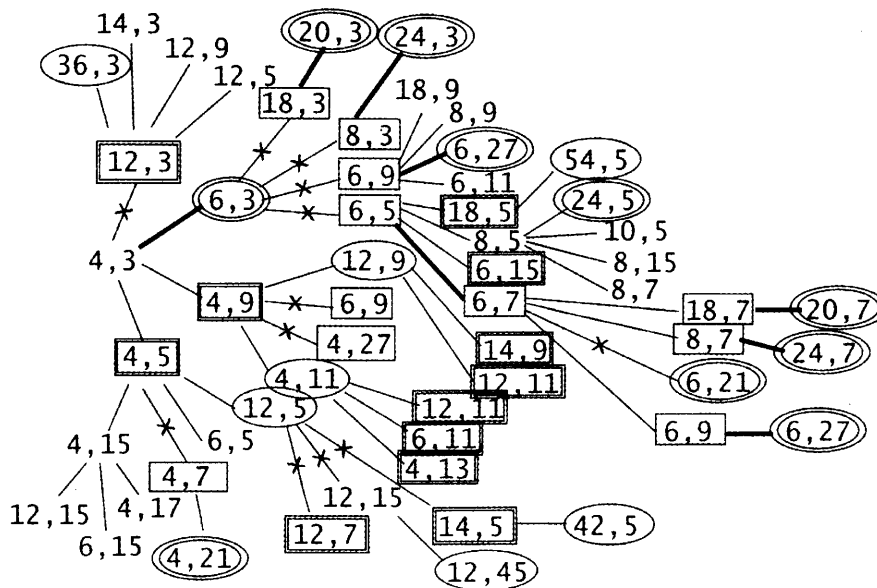


Рисунок 7.5. Дерево игры. Варианты пятого хода

Заметим, что многие рассмотренные нами варианты хода оказались в результате ненужными оттого, что соседние с ними варианты приводили к победе. Вы можете попытаться найти для себя способ сразу «видеть» только те ветви дерева, которые нужны для решения. Нам представляется это излишним. На выписывание этих «лишних» вариантов уходит не так уж много времени, а потеря хотя бы одного критичного варианта чревата ошибкой всего решения.

Таким образом, мы решили задачу.

Как вы думаете, сколько баллов (из трех возможных) мы заработали за это задание?

Надеемся, вы уже прочитали критерии оценивания этой задачи и поняли, что пока мы заработали ноль баллов. Если вы еще не прочитали критерии оценивания, внимательно прочитайте их сейчас:

Указания по оцениванию	Баллы
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры).	3
Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности.	2

Указания по оцениванию	Баллы
При наличии в представленном решении одного из пунктов: 1. Правильно указан выигрывающий игрок и его первый ход, рассмотрены все возможные ответы второго игрока, но неверно определены дальнейшие действия. 2. Правильно указан выигрывающий игрок и его первый ход, но описание выигрышной стратегии неполно и рассмотрены несколько (больше одного, но не все) вариантов ответов второго игрока.	1
Задание не выполнено или в представленном решении полностью отсутствует описание элементов выигрышной стратегии и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока).	0

Мало решить задачу. Нужно еще правильным образом оформить ее решение и ответ.

Вы уже прочитали, что все три балла можно получить только в том случае, если приведено доказательство правильности выбранного решения.

Полученное нами дерево игры, с одной стороны, полностью отображает стратегию игры. С другой стороны, при нужном описании его также можно будет считать доказательством правильности решения. Мы, однако, предлагаем только решать задачу таким способом, а вот ответ (и доказательство) записывать так, как это приведено в критериях оценивания этой задачи, по которым вашу работу будут оценивать эксперты.

Это связано с тем, что у эксперта при проверке в образце задание будет оформлено именно так. И если вы приведете решение, полностью соответствующее этому образцу, у эксперта будет гораздо меньше сомнений, должен ли он ваш ответ засчитать как верное доказательство правильности. Он просто убедится, что вы слово в слово привели образцовый ответ и поставит вам ваши заслуженные 3 балла.

Рекомендуемый образец оформления ответа:

Выигрывает Петя, своим первым ходом он должен добавить 2 камня к первой кучке. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучках соответственно.

Стартовая позиция	I ход (Петя), выигрышный	II ход (Вани), все варианты хода	III ход (Петя), выигрышные ходы	IV ход (Вани), все варианты хода	V ход (Петя), выигрышные ходы		
4,3	6,3	18,3	20,3	Выигрыш Пети			
		8,3	24,3				
		6,9	6,27				
		6,5	6,7	18,7	20,7	Выигрыш Пети	
				8,7	24,7	Выигрыш Пети	
				6,21	6,27	Выигрыш Пети	
				6,9	6,27	Выигрыш Пети	

Таблица содержит все возможные варианты ходов второго игрока (Вани). Из нее видно, что при любом ответе второго игрока у первого (Петя) имеется ход, приводящий к победе.

Заметим, что приведенный образец оформления отчета обладает сразу тремя достоинствами: во-первых, это самый короткий способ доказательства; во-вторых, он содержит в себе также и стратегию выигрышной игры Пети; в третьих (и это самое важное), именно такой способ оформления решения ожидают увидеть эксперты при проверке.

Обратите внимание! Мы рекомендуем выучить текст наизусть и привести именно эти фразы. В таблице обязательно подписать столбцы, указав для проигрывающего игрока «все варианты хода», а для выигрывающего игрока «выигрышные ходы». После таблицы ОБЯЗАТЕЛЬНО приведите эти две фразы, которые позволяют считать, что вы действительно доказали правильность решения.

Практическая часть

7.2. Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 4, а во второй — 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то кучке, или добавляет 2 камня в какую-то кучку. Игра завершается в тот момент, когда количество камней в одной из кучек становится не менее 19. Если в момент завершения игры общее число камней в двух кучках не менее 33, то выиграл Ваня, в противном случае — Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Ответ: _____.

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход	

7.3. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу 1 камень или удвоить количество камней в куче. Например, имея ку-

чу из 7 камней, за один ход можно получить кучу из 8 или 14 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 30. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 30 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 29$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) При каких значениях числа S Петя может выиграть первым ходом? Укажите все такие значения.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения S , при которых у Пети есть выигрышная стратегия, причем Петя не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.

3. Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

Решение и рекомендации по оформлению.

В этой задаче (к счастью) у Вас подробно, по пунктам, спрашивают ответ. Не ленитесь, так же подробно, по каждому пункту отвечайте развернуто и понятно.

Курсивом выделен образец ответа.

1а) Нужно найти такие значения S , при которых можно одним ходом выиграть. Очевидно, что проще всего выиграть ходом «удвоить количество камней». Найдем такие значения, удвоив которые можно сразу получить число 30 и больше. Это $30/2 = 15$. Значит, при любых допустимых $S \geq 15$ можно одним ходом выиграть. Образец оформления:

1а) При $S = 15 \dots 29$. Во всех этих случаях Петя должен удвоить количество камней в куче и выиграть. При значениях $S < 16$ невозможно одним ходом (+1 или *2) получить больше 30 камней.

Обязательно укажите: диапазон верных значений, как нужно ходить, чтобы выиграть, а также почему для остальных S это не верно. Заметьте, что указание хода, который нужно сделать, является обязательным, несмотря на то, что этот ход (удвоить количество камней) очевиден, его нужно обязательно явно указать.

Также заметим, что для значения 29 не обязательно отдельно и дополнительно указывать, что выиграть можно также ходом «+1». Одного выигрышного хода более чем достаточно.

(Если Вы его укажете, это даже, скорее всего, немного расстроит эксперта, потому что, как указано в условии, «выигрышная стратегия» — это когда для каждой ситуации указан выигрышный ход. Один! Не нужно приводить все варианты выигрышных ходов.)

16) Нужно найти такое значение S , из которого, куда бы ни пошел, обязательно попадешь в диапазон, найденный в пункте 1а. Обычно это число, непосредственно на 1 меньшее, чем левая граница этого диапазона. В данном случае это 14. Образец оформления:

16) При $S = 14$. Куда бы ни пошел Петя ($14 + 1 = 15$ или $14 * 2 = 28$), Ваня удвоит число камней в куче и выиграет ($15 * 2 = 30$, $28 * 2 = 56$).

Обязательно укажите:

- 1) верное значение,
- 2) какие варианты хода Пети,
- 3) как должен ходить в ответ Ваня.

Не нужно указывать несколько вариантов выигрышного хода, даже если они имеются. Указывайте только один. При этом тот, который совпадает с вариантом хода Вани при другом ходе Пети. Так проще, короче и понятнее.

2. Нужно рассмотреть «предыдущий ход» относительно позиции, указанной в пункте 16. То есть нужно указать такие позиции, из которых можно одним ходом получить число, указанное в пункте 16. В данном случае 14 можно получить, прибавив 1 к 13 или умножив на 2 число 7. Образец оформления:

2. $S = 13$ или $S = 7$. В обоих случаях Петя может получить в куче 14 камней ($13 + 1 = 14$, $7 * 2 = 14$). При любом ответном ходе Вани ($14 + 1 = 15$ или $14 * 2 = 28$) Петя должен удвоить число камней в куче и выиграть ($15 * 2 = 30$, $28 * 2 = 56$).

Несмотря на то, что с каждым следующим пунктом кажется, что можно каждый раз ссылаться на предыдущий пункт, при этом каждый раз меняется выигрывающий игрок. Поэтому проще заново описать стратегию, чем потом оправдываться на апелляции, что Вы имели ввиду, когда пытались сэкономить пару строк, а получили не очень понятный текст ответа. Поэтому повторяем еще раз.

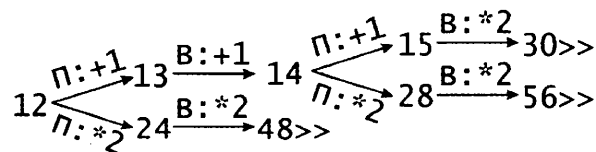
Обязательно укажите:

- 1) верные значения,
- 2) для каждого из них: как должен ходить Петя (и сколько камней получится в куче),
- 3) как может ходить в ответ Ваня (и сколько при этом будет в куче камней),
- 4) как должен в ответ ходить Петя.

3. Здесь нас просят снова рассмотреть «предыдущий ход» относительно позиции 2. То есть указать такую позицию, из которой куда бы ни ходил Петя, он либо попадет в диапазон пункта 1а (выигрыш Вани за 1 ход), либо попадет в одну из позиций пункта 2 (выигрыш Вани за 2 хода). Рекомендуем рассмотреть такие позиции, из которых можно получить за один ход позиции из пункта 2.

В данном случае, так как возможные ходы «+1» или «*2», а в пункте 2 оба числа (7 и 13) нечетные, достаточно рассмотреть числа 6 и 12 ($6 + 1 = 7$ и $12 + 1 = 13$). В обоих случаях рекомендуем построить дерево возможных ходов и в одном из них убедиться, что Ваня выигрывает за 1 или за 2 хода. Его и записать в ответ.

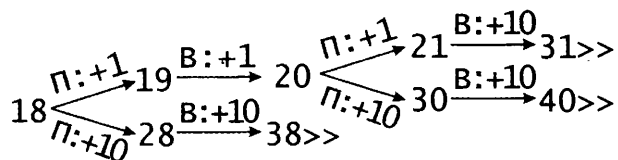
В данном случае при $S = 6$ Петя может ходить в 7 или в 12, и Ваня ни в одной из этих позиций не может выиграть в один ход (как требуется по условию). Поэтому можно сразу рассматривать $S = 12$. Строим дерево:



Убеждаемся, что все получается так, как нужно, и записываем ответ:

3) $S = 12$. Если Петя удвоит число камней в куче ($12 * 2 = 24$), Ваня тоже должен удвоить число камней в куче и выиграть. Если Петя добавит 1 камень в кучу ($12 + 1 = 13$), Ваня также должен добавить 1 камень в кучу ($13 + 1 = 14$).

В ответ на любой следующий ход Пети ($14 + 1 = 15$ или $14 * 2 = 28$) Ваня должен удвоить число камней в куче и выиграть ($15 * 2 = 30$, $28 * 2 = 56$). Рассмотрим дерево игры:



В этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня, — только ход, соответствующий стратегии, которую выбрал Ваня.

Обратите внимание на то, что рекомендуется здесь написать:

- 1) опишите словами стратегию выигрыша Вани,
- 2) нарисуйте дерево игры. На ребрах дерева указывайте, кто сейчас ходит и как,
- 3) обязательно приведите фразу, указанную после дерева («В этом дереве...»). Выучите этот текст и напишите его в чистовик при оформлении этой задачи. Без этого текста Вам могут не засчитать правильное решение пункта 3.

7.4. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 31 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 30$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) При каких значениях числа S Петя может выиграть первым ходом? Укажите все такие значения.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения S , при которых у Пети есть выигрышная стратегия, причем Петя не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.

3. Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

7.5. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество

камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырех позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 55. Победителем считается игрок, сделавший последний ход, то есть первым получивший такую позицию, что в кучах всего будет 55 или больше камней.

В начальный момент в первой куче было 5 камней, во второй куче — S камней; $1 \leq S \leq 49$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

Задание 1

а) Укажите все такие значения числа S , при которых Петя может выиграть за один ход, и соответствующие выигрышающие ходы. Если при некотором значении S Петя может выиграть несколькими способами, достаточно указать один выигрышающий ход.

б) Сколько существует значений S , при которых Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом?

Задание 2

Укажите такое значение S , при котором у Пети есть выигрышная стратегия, причем одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для указанного значения S опишите выигрышную стратегию Пети.

Задание 3

Укажите значение S , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На ребрах дерева указывайте ходы, в узлах указывайте позиции.

В заданиях 2 и 3 достаточно указать одно значение S и объяснить, почему это значение удовлетворяет условию соответствующего задания.

Решение и рекомендации по оформлению.

1а. В первой куче 5 камней. Выигрыш при 55 камнях. Значит, во второй куче нужно получить 50 и более камней. Ходом удвоения можно получить 50 и более камней, имея во второй куче хотя бы 25 камней ($25 \cdot 2 = 50$). Значит, наш ответ:

$S = 25 \dots 49$. Петя должен удвоить число камней во второй куче.

1б. Единственный вариант, который может нас устроить, $S = 24$. Проверим его.

Из позиции (5, 24) можно получить позиции (6, 24), (10, 24), (5, 25), (5, 48). Среди этих позиций из позиции (6, 24) нельзя выиграть одним ходом (самый «максимальный» по сумме

камней ход $24 \cdot 2$ даст всего лишь $(6, 48) = 54$ камня в двух кучах). Значит, этот вариант $S = 24$ не подходит. При $S < 24$ очевидно, что существуют ходы, прибавляющие 1 ко второй куче, которые приводят нас последовательно к случаям, которые не подходят. Значит, ответ на вопрос 16 — 0 (нет таких S). Рекомендуем привести в ответе доказательство правильности этого ответа. Например, так:

Единственный подходящий вариант ($S = 24$) не подходит, так как при первом ходе Пети «+1 к первой куче» получается ситуация $(6, 24)$, из которой нельзя выиграть одним ходом. Числа, меньшие 24, тоже не подходят (при $S = 23$ добавление 1 камня во вторую кучу приводит к $(5, 24)$, из которой не выиграть). Аналогично, можно такое показать для каждого меньшего S .

2. Только что рассмотренный случай $S = 24$ приводит нас к ответу на это задание. Мы заметили, что ситуация $(6, 24)$ является выигрышной. Проверим, проигрышной ли она. Рассмотрим, как может ходить игрок из позиции $(6, 24)$. Это позиции $(7, 24)$, $(12, 24)$, $(6, 25)$, $(6, 48)$. Легко видно, что во всех этих случаях удвоение второй кучи дает в сумме в кучах не менее 55 камней. То есть, позиция $(6, 24)$ проигрышная. Значит, именно в нее нужно стараться «загнать» соперника. Ответ:

$S = 24$. Первым ходом Петя должен добавить 1 камень в первую кучу (получится $(6, 24)$). При любом ответном ходе Вани Петя должен удвоить вторую кучу и выиграть.

3. Воспользуемся выводами, сделанными нами в задании 2. Позиция $(5, 24)$ является выигрышной, а позиция $(6, 24)$ — проигрышной. Из какой позиции, куда бы ни ходил игрок, он попадет в позицию $(5, 24)$? Например, из $(5, 23)$. Проверим эту позицию.

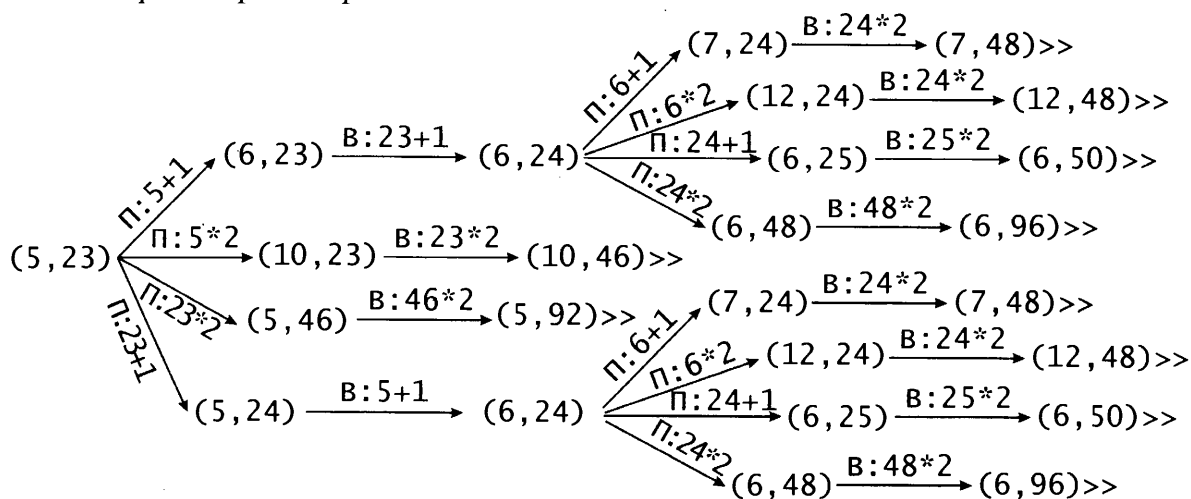
Из $(5, 23)$ одним ходом можно попасть в позиции $(6, 23)$, $(5, 24)$, $(10, 23)$ и $(5, 46)$. Из позиций $(6, 23)$ и $(5, 24)$ можно одним ходом (+1) получить позицию $(6, 24)$, которую мы уже определили как проигрышную.

А из позиций $(10, 23)$ и $(5, 46)$ можно одним ходом выиграть, удвоив число камней во второй куче.

Значит, наше предположение верно. Осталось только оформить ответ:

$S = 23$.

Рассмотрим дерево игры:



В дереве рассмотрены все возможные ходы Пети и ответные ходы Вани, соответствующие выигрышной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.

7.6. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать $(10, 7)$. Тогда за один ход можно получить любую из четырех позиций: $(11, 7)$, $(20, 7)$, $(10, 8)$, $(10, 14)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 65. Победителем считается игрок, сделавший последний ход, то есть первым получивший такую позицию, что в кучах всего будет 65 или больше камней.

В начальный момент в первой куче было 7 камней, во второй куче — S камней; $1 \leq S \leq 57$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

Задание 1

а) Укажите все такие значения числа S , при которых Петя может выиграть за один ход, и соответствующие выигрышающие ходы. Если при некотором значении S Петя может выиграть несколькими способами, достаточно указать один выигрышающий ход.

б) Сколько существует значений S , при которых Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом?

Задание 2

Укажите такое значение S , при котором у Пети есть выигрышная стратегия, причем одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для указанного значения S опишите выигрышную стратегию Пети.

Задание 3

Укажите значение S , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На ребрах дерева указывайте ходы, в узлах указывайте позиции.

В заданиях 2 и 3 достаточно указать одно значение S и объяснить, почему это значение удовлетворяет условию соответствующего задания.

7.7. Два игрока, Паша и Витя, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 кам-

ней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырех позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 47. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 47 или больше камней.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (3, 22) и (1, 23) выигрышная стратегия есть у Паши. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций (6, 20), (8, 19) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (5, 20), (7, 19), (8, 18) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (7, 18) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

Решение и рекомендации по оформлению.

Решаем задания по порядку, начиная с первого.

Задание 1.

Проанализируем начальные позиции первого задания.

Из позиции (6,20) можно получить (7,20), (12,20), (6,21), (6,40). Во всех случаях сумма камней в кучах меньше 47, однако удвоение второй кучи приводит к выигрышу (в сумме будет не менее 47 камней). Значит, позиция (6,20) — проигрышная и выигрышную стратегию имеет Витя. Он выигрывает своим первым ходом.

Из позиции (8,19) можно получить (9,19), (16,19), (8,20), (8,38). Во всех этих случаях сумма камней в кучах также меньше 47 и удвоение второй кучи приводит к выигрышу. Значит, позиция (8,19) — также проигрышная и выигрышную стратегию имеет Витя. Он выигрывает своим первым ходом.

Что нужно записать в ответ?

Нас просят указать сразу 4 вещи для каждой из 2-х позиций (то есть всего 8):

- 1) указать, кто из игроков имеет выигрышную стратегию,
- 2) описать выигрышную стратегию,
- 3) объяснить (доказать), почему эта стратегия ведет к выигрышу,

4) указать, какое наибольшее число ходов может потребоваться победителю.

Чтобы ответ был засчитан как доказательство, нужно обязательно привести все возможные ходы проигрывающего игрока (и позиции, в которых окажется игра в этот момент) и в каждом случае указать, как должен ходить выигрывающий игрок.

Например, можно оформить ответ так, как указано выше курсивом.

Задание 2.

Желательно воспользоваться тем результатом, который мы получили при решении задания 1.

А именно тем, что позиции (6,20) и (8,19) — проигрышные.

Нетрудно видеть, что из позиций задания 2 можно легко одним ходом получить позиции задания 1. Значит, они — выигрышные. Запишем ответ:

Для каждой из рассматриваемых позиций существует ход, который приводит в выигрышные позиции, рассмотренные в задании 1.

$(5,20) \rightarrow (6,20)$, $(7,19) \rightarrow (8,19)$, $(8,18) \rightarrow (8,19)$.

Значит, эти позиции — выигрышные. Выигрышную стратегию имеет Паша. Его первый ход нами указан. Паша выигрывает на своем втором ходу. Игра из позиций (6,20) и (8,19) рассмотрена в задании 1.

Задание 3.

Рассматриваем указанную позицию (7,18). Из нее можно получить позиции (8,18), (7,19), (14,18), (7,36). Позиции (8,18) и (7,19) рассмотрены в задании 2 как выигрышные.

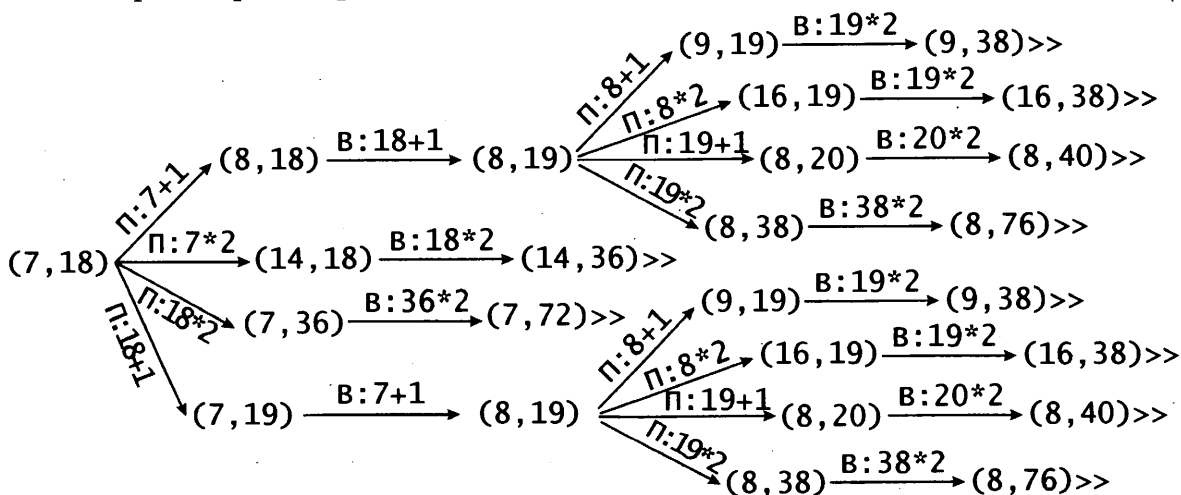
Позиции (14,18) и (7,36) выигрышные, потому что в них удвоение второй кучи дает в кучах более 47 камней.

Запишем ответ:

В позиции (7,18) выигрышную стратегию имеет Витя. Если Паша первым ходом пойдет в (8,18) или (7,19), Витя пойдет в (8,19) и выиграет своим вторым ходом. Стратегия его игры описана в заданиях 1 и 2.

Если же Паша первым ходом пойдет в (14,18) или (7,36), Витя выиграет своим вторым ходом, удвоив количество камней во второй куче.

Рассмотрим дерево игры:



В этом дереве рассмотрены все возможные ходы Паши и соответствующие им ходы Вити согласно выигрышной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

7.8. Два игрока, Паша и Витя, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать $(10, 7)$. Тогда за один ход можно получить любую из четырех позиций: $(11, 7)$, $(20, 7)$, $(10, 8)$, $(10, 14)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 63. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 63 или больше камней.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях $(3, 30)$ и $(1, 31)$ выигрышная стратегия есть у Паши. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций $(10, 26)$, $(12, 25)$ укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций $(6, 25)$, $(9, 26)$, $(10, 25)$ укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции $(9, 25)$ укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

8. ФАЙЛОВАЯ СИСТЕМА КОМПЬЮТЕРА

Основные понятия

Для удобства использования данные на компьютере хранятся в виде файлов. *Файлом* называется поименованная совокупность данных, хранящихся на внутреннем (жесткий диск, флэш-диск) или внешнем (CD-диск, DVD-диск, дискета, флэш-карта, карта памяти, магнитная лента и др.) носителе информации.

Имя файла обычно состоит из собственно имени и так называемого *расширения*, служащего для определения типа данных, хранящихся в файле (типа файла). Имя файла и его расширение разделяются точкой.

Кроме имени, файл обычно имеет *атрибуты* (дата создания, дата изменения, пользователь, права доступа и т.д.).

Файловая система позволяет организовывать файлы в иерархические древовидные *каталоги*. Каталог самого верхнего уровня называется *корневым каталогом*. Во многих операционных системах (в том числе семейства Windows) корневой каталог обозначается именем логического источника данных — латинской буквой с двоеточием. Буквы А и В обозначают дисководы для дискет, С — жесткий диск компьютера, остальные буквы — внутренние носители данных или внешние, подключенные к компьютеру, например по интерфейсу USB или по сети.

Остальные каталоги делятся на *системные* и *пользовательские*. Системные каталоги имеют служебные имена, назначенные операционной системой при установке. Имена пользовательских каталогов задаются пользователем по его усмотрению.

Пользователь может передвигаться по каталогам с помощью *программ навигации* (Проводник, FarManager и др.) или с помощью специальных команд операционной системы. Тот каталог, в котором пользователь находится в настоящий момент, называется *текущим каталогом*.

Последовательность имен каталогов, проходимых от некоторого каталога до нужного файла, называется *путем* к этому *файлу*. Имена каталогов в этой последовательности принято разделять знаком «\».

Различают абсолютное (полное) и относительное имя файла. *Абсолютное имя* складывается из пути к файлу от корневого каталога и имени файла. *Относительное имя* — из пути к файлу из текущего каталога и имени файла. Если файл находится в текущем каталоге, то его относительное имя есть просто имя файла.

Пример. Пусть имя файла E:\SCHOOL\USER\DOC\user.txt

Это полное (абсолютное) имя файла, поскольку оно начинается с имени корневого каталога E:. Путь к файлу — E:\SCHOOL\USER\DOC\. Само имя файла — user.txt, где txt — расширение, указывающее на то, что файл должен содержать текстовые данные.

Для групповых операций с файлами (поиск, удаление, перемещение) используются *маски* имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

символ «?» (вопросительный знак) означает ровно один произвольный символ;

символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Примеры.

Маска для выделения всех файлов в текущем каталоге: * . *

Маска для выделения файлов с расширением .txt: * .txt

Маска для выделения файлов, имена которых начинаются с буквы s и имеют односимвольное расширение: s * .?.

Практическая часть

8.1. Выпишите имя корневого каталога, путь к файлу, его расширение:

а) В:\TXT\DOC\MY.CPP

Ответ: _____.

б) Y:\Y\Z\Z.Y

Ответ: _____.

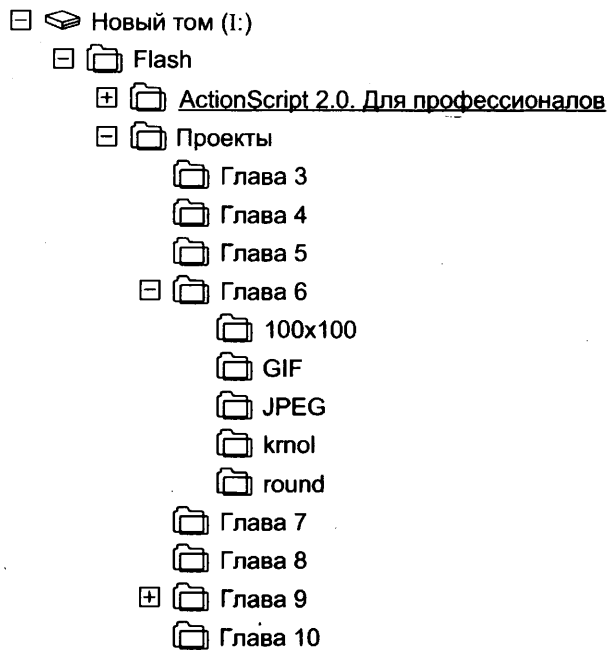
в) C:\Мои документы\фото\2007\портрет.jpg

Ответ: _____.

г) C:\текст.doc

Ответ: _____.

8.2. На рисунке приведен фрагмент файловой системы компьютера.



а) Запишите полный путь к файлу 333.gif, находящемуся в каталоге GIF.

Ответ: _____.

б) Запишите путь из каталога Проекты к файлу 333.gif, находящемуся в каталоге GIF.

Ответ: _____.

в) Запишите путь из каталога Глава 6 к файлу 333.gif, находящемуся в каталоге GIF.

Ответ: _____.

г) Запишите полный путь к файлу test.swf, находящемуся в каталоге Проекты.

Ответ: _____.

Пример. Какие файлы будут выбраны из текущего каталога по маске d * c.?p

Решение.

По указанной маске будут выбраны из текущего каталога все файлы, имена которых начинаются на d и заканчиваются на c и одновременно имеют двухбуквенное расширение, заканчивающееся на p.

Пример. Составьте маску, по которой из текущего каталога будут выбраны все файлы, имена которых начинаются на букву «F» с расширением, состоящим ровно из трех символов.

Решение.

Составим отдельно маски для расширения и собственно имени файла.

Поскольку указано, что расширение состоит ровно из трех символов, для него необходимо использовать маску «???». Символ «*» в данном случае использовать нельзя, так как он не накладывает ограничений на длину подставляемой последовательности символов. Напротив, поскольку для имени файла ограничения на длину не предусмотрены — маска для имени будет «F *».

Объединяя маски для имени и расширения и не забывая про разделитель имени и расширения — точку, получаем ответ: F * .???

Пример. Определите, какое из указанных имен файлов удовлетворяет маске ?cr * m.d?c

- 1) crim.doc 2) acrim.doc 3) ocrm.dtc 4) ocrm.dc

Решение.

Проверим последовательно, какие имена файлов удовлетворяют указанной маске:

- 1) crim.doc — не подходит, так как символу «?» в имени файла ничего не соответствует, а это недопустимо;
- 2) acrim.doc — не подходит, поскольку в имени файла есть лишний символ «s», не соответствующий маске;
- 3) ocrm.dtc — подходит, символу «?» маски в имени файла соответствует ровно одна буква «o», а в расширении «t». Символу «*» маски в имени файла соответствует пустая последовательность символов, что допускается.

Для полноты решения проверим четвертый вариант.

- 4) ocrm.dc — не подходит, так как символу «?» в расширении файла ничто не соответствует, что недопустимо.

Ответ: 3.

8.3. В текущем каталоге имеются файлы:

- 1) zzz.cpp 2) zzz.cpprc 3) z.c 4) 1.cpp 5) 123.pc

а) Какие файлы будут выбраны из текущего каталога по маске * .crr

Ответ: _____.

б) Какие файлы будут выбраны из текущего каталога по маске ?.crr

Ответ: _____.

в) Какие файлы будут выбраны из текущего каталога по маске * z * .c?r

Ответ: _____.

г) Какие файлы будут выбраны из текущего каталога по маске * . * c

Ответ: _____.

8.4.

а) Составьте маску, по которой из текущего каталога будут выбраны все файлы с расширением .doc, в именах которых встречается буква «я».

Ответ: _____.

б) Составьте маску, по которой из текущего каталога будут выбраны все файлы с расширением .doc, имена которых начинаются и заканчиваются на букву «я».

Ответ: _____.

в) Составьте маску, по которой из текущего каталога будут выбраны все файлы, имена которых начинаются на букву «z» с расширением, состоящим ровно из двух символов.

Ответ: _____.

г) Составьте маску, по которой из текущего каталога будут выбраны все файлы с расширением .txt, имена которых включают идущие подряд цифры 2009.

Ответ: _____.

Пример. В каталоге находится 5 файлов:

boom.pas
bloom.ppt
loom.pas
bottom.pdf
bom.pps

Определите, по какой из масок будет выбрана указанная группа файлов:

boom.pas
bloom.ppt
bottom.pdf

1) ?oo*m.p*

2) b*om.p?*

3) *o*om.p*?

4) b*o*om.p??

Решение.

Найдем имена исходных файлов, которые не были отобраны по маске: loom.pas, bom.pps. Проверим последовательно каждую маску на то, чтобы все выбранные файлы ей удовлетворяли, а при этом невыбранные файлы ей не удовлетворяли:

- 1) ?oo*m.p* — не подходит, так как символ «?» в начале имени файла требует, чтобы перед «oo» находился ровно один символ, а этому условию не удовлетворяет выбранный файл bloom.ppt;
- 2) b*om.p?* — не подходит, поскольку этой маске удовлетворяют не только выбранные файлы, но и файл bom.pps, который не должен быть отобран;
- 3) *o*om.p*? — не подходит, поскольку этой маске удовлетворяют не только выбранные файлы, но и файл loom.pas, который не должен быть отобран.

Для полноты решения проверим четвертый вариант.

- 4) b*o*om.p?? — подходит, все отобранные файлы соответствуют этой маске; loom.pas ей не соответствует, потому что его первая буква «l», а маска требует «b»; bom.pps ей не соответствует, потому что в нем только одна буква «o», а маска требует двух.

Ответ: 4.

Задания формата ЕГЭ

8.5.

- а) В некотором каталоге хранился файл Лермонтов.doc. После того как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл Лермонтов.doc, полное имя файла стало F:\Россия\Поэты\XIX\Лермонтов.doc. Каково имя вновь созданного каталога?

- 1) Поэты
- 2) F:\Россия
- 3) XIX
- 4) Россия

- б) В некотором каталоге хранился файл Задачи.doc. После того как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл Задачи.doc, полное имя файла стало G:\Математика\9\Алгебра\Задачи.doc. Каково имя вновь созданного каталога?

- 1) Алгебра
- 2) Математика
- 3) 9
- 4) G:\Математика

- в) Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги Задачи, 9, Алгебра, C:\, Геометрия, 11. Каково полное имя каталога, в котором оказался пользователь?

Примечание: при каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше.

- 1) 11
- 2) C:\ Алгебра\9\Задачи
- 3) C:\Геометрия\11
- 4) Геометрия\11

- г) Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги Задачи, Геометрия, В:\, Алгебра, Уравнения, Задачи. Каково полное имя каталога, в котором оказался пользователь?

Примечание: при каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше.

- 1) Алгебра\Уравнения\Задачи
- 2) В:\ Алгебра\Уравнения\Задачи
- 3) В:\Геометрия\Задачи
- 4) Геометрия\Задачи

8.6.

- а) Определите, какое из указанных имен файлов удовлетворяет маске: hel?lo.c??

- 1) hello.cpp
- 2) helolo.c
- 3) helolo.cnt
- 4) helsslo.cpp

- б) Определите, какое из указанных имен файлов удовлетворяет маске: he * o.c *
- 1) helol.cpp 2) he.oc 3) helolo.pcc 4) herro.c
- в) Определите, какое из указанных имен файлов удовлетворяет маске: ?com * .d?c
- 1) com.doc 2) dcoma.dc 3) dcom.dotc 4) dcoma.djc
- г) Определите, какое из указанных имен файлов удовлетворяет маске: * com?.d * c
- 1) com.doc 2) dcoma.dc 3) dcom.dotc 4) dcoma.cdc

8.7.

- а) Укажите, какой маске удовлетворяют следующие файлы:
com.doc
dcoma.doc
dcom.dotc
dcoma.cdc
- 1) *com?.d* 2) *com*.d?? 3) *com*.*d* 4) ?com?.*
- б) Укажите, какой маске удовлетворяют следующие файлы:
123.tex
12.txt
1234.tex
123.txt
- 1) 12*.tx* 2) *12*.t?x? 3) 12??.*x* 4) 12*.t??
- в) Укажите, какой маске удовлетворяют следующие файлы:
abc.txt
abd.tif
abcd.txt
abc.tif
- 1) ???t?? 2) ab*?.*t?? 3) ab??t* 4) abc*.*
- г) Укажите, какой маске удовлетворяют следующие файлы:
xyz.hlp
xyt.htm
xyz.htm
xtz.hlp
- 1) xy*.*?? 2) *.*h* 3) x??h?? 4) x*z.h*

8.8.

- а) Некий файл хранится по следующему адресу:
D:\VASYA\PICTURES\FACE.JPG.
- Укажите заведомо неверное утверждение.

- 1) Из папки VASYA путь к файлу можно указать так: PICTURES\FACE.JPG
 - 2) Этот файл содержит изображение черного треугольника на синем фоне
 - 3) Полное имя этого файла: FACE.JPG
 - 4) Этот файл хранится не в корне диска D:
- б) Некий файл хранится по следующему адресу:
A:\MATUSYA\HOMEWORK\WINTER.DOC
Укажите заведомо неверное утверждение.
- 1) Полное имя этого файла: A:\MATUSYA\HOMEWORK\WINTER.DOC
 - 2) Из папки MATUSYA путь к файлу можно указать так: HOMEWORK\WINTER.DO
 - 3) Этот файл хранится в корне диска A:
 - 4) Этот файл содержит таблицу расписания поездов
- в) Определите, какое из указанных имен файлов удовлетворяет маске: ??prive * t.p? *
- 1) Aprivet.pas
 - 2) z1privet.past
 - 3) privert.p3
 - 4) doprivest.p
- г) Определите, какое из указанных имен файлов удовлетворяет маске: ?info * rm.d * ?
- 1) 12info7rm.deep
 - 2) 11inform.doc
 - 3) 4info5rm.d
 - 4) 7inform.dp

8.9.

- а) В каталоге находится 5 файлов:
 sound.tif
 son.txt
 spoon.ttf
 slon.tif
 pound.txt
- Определите, по какой из масок будет выбрана указанная группа файлов:
 son.txt
 spoon.ttf
 slon.tif
- 1) *n?.t* 2) s*n.t?? 3) *s*.t* 4) *o*.???
- б) В каталоге находится 5 файлов:
 sound.tif
 son.txt
 sun.ttf
 slon.tif
 pound.txt

Определите, по какой из масок будет выбрана указанная группа файлов:

son.txt

sun.ttf

pound.txt

1) ?*n*.t*

2) s*.t??

3) s*n*.??*

4) *n*.?*t*

в) В каталоге находится 5 файлов:

sound.tif

son.txt

sun.ttf

slon.tif

pound.txt

Определите, по какой из масок будет выбрана указанная группа файлов:

sound.tif

sun.ttf

pound.txt

1) *un?.t??

2) *u*.?t*

3) ?*un*.t*

4) ?*n*.?t*

г) В каталоге находится 5 файлов:

sound.tif

son.txt

spoon.ttf

slon.tif

pound.txt

Определите, по какой из масок будет выбрана указанная группа файлов:

sound.tif

spoon.ttf

slon.tif

1) s*o*n.???

2) s*.*t??

3) s?o*.t?f

4) ?*o*.t*f

9. ЭЛЕКТРОННЫЕ ТАБЛИЦЫ

Основные понятия

Так как одним из самых удобных способов представления информации является таблица, обработке таблиц посвящена отдельная категория программных продуктов — *электронные таблицы*. Электронная таблица состоит из строк и столбцов, на пересечении которых находятся ячейки. Каждая ячейка может хранить один из нескольких видов информации: текстовую, числовую, дату/время или формулу.

В электронных таблицах принято обозначать столбцы буквами латинского алфавита, а строки — числами. Таким образом, для обозначения ячейки нужно указать номер столбца и номер строки этой ячейки. Например, D7. Это называется *адресом ячейки*.

Для обработки информации, содержащейся в ячейках, используют формулы.

Формулы в электронных таблицах начинаются с символа равно (=). В качестве аргументов в формулах можно использовать числа, элементы текста и адреса других ячеек. Эти аргументы можно связывать различными алгебраическими операциями, операциями отношения и другими.

Для удобства в электронных таблицах введены различные функции. Самые широкоиспользуемые — вычисление суммы (СУММ) и вычисление среднего арифметического значения (СРЗНАЧ).

Для перечисления адресов ячеек, над которыми должна быть осуществлена функция, используют знак «;» (точка с запятой). Для указания прямоугольного блока ячеек используется обозначение *диапазона*: адрес верхней левой и правой нижней ячеек блока, с символом «:» (двоеточие) между ними.

При копировании формул из одной ячейки в другую действует *автоиндексация*. Это значит, что адреса ячеек при копировании будут изменены в соответствии с расстоянием, на которое осуществляется копирование. То есть если копируется формула из ячейки на 3 столбца вправо и на 2 строки вверх, то во всех адресах, которые присутствуют в формуле, номера столбцов будут «увеличены» на 3, а номера строк уменьшены на 2.

Чтобы такого эффекта не происходило, перед теми элементами адреса (номером столбца и/или номером строки), которые не должны измениться при копировании, нужно ставить знак доллара (\$). Это называется *абсолютная адресация* ячейки. В противоположность ей, отсутствие знака доллара в адресе называется *относительной адресацией*.

Практическая часть

9.1. Укажите одно или несколько правильных утверждений. Буквы, соответствующие правильным утверждениям, обведите, остальные — зачеркните.

Для электронных таблиц справедливо:

- а) электронная таблица состоит из строк и столбцов, на пересечениях которых находятся ячейки;
- б) строки и столбцы электронной таблицы пронумерованы. Строки обычно нумеруются последовательными натуральными числами. Столбцы обычно нумеруются буквами латинского алфавита, от А и дальше;
- в) каждая ячейка электронной таблицы имеет уникальный адрес, состоящий из номера столбца и номера строки (например, А1, С14, ВD23).

9.2. Укажите один или несколько правильных ответов. Буквы, соответствующие правильным ответам, обведите, остальные — зачеркните.

Какая информация может храниться в ячейке электронной таблицы?

- | | |
|----------------------------|----------------------|
| а) Целое число | д) Время/дата |
| б) Вещественное число | е) Звук |
| в) Графическое изображение | ж) Вложенная таблица |
| г) Строка символов | з) Формула |

9.3. Для расчета значения ячейки на основании значения другой ячейки (ячеек) в электронных таблицах используются формулы. Укажите выражения, которые электронная таблица будет считать формулами и вычислит. Буквы, соответствующие правильным ответам, обведите, остальные — зачеркните:

- | | |
|----------------|------------------|
| а) $A1 + 3$ | е) $A1 = A3$ |
| б) $= 5$ | ж) $A1 + A3$ |
| в) $= A1 - 2$ | з) $= A1 = A2$ |
| г) $A1 < A3$ | и) $= A(12 + 4)$ |
| д) $= 53 - 10$ | к) $= A3 > A2$ |

9.4. Для осуществления операций над константами и значениями ячеек в электронной таблице используются операторы. Сопоставьте каждому оператору (слева) действие, которое этот оператор осуществляет (справа). Впишите рядом с буквой операции число, соответствующее смыслу этой операции:

- | | |
|---------|----------------------------------------------------------------|
| а) $>$ | 1) возведение в степень (первый аргумент в степени второго) |
| б) $-$ | 2) сравнение аргументов (если первый меньше или равен второму) |
| в) $*$ | 3) сравнение аргументов (если первый больше второго) |
| г) $=$ | 4) сложение аргументов |
| д) $^$ | 5) умножение аргументов |
| е) $\&$ | 6) деление первого аргумента на второй |
| ж) $+$ | 7) сравнение аргументов (если первый больше или равен второму) |
| з) $>=$ | 8) сравнение аргументов (если аргументы не равны) |
| и) $/$ | 9) вычитание второго аргумента из первого |
| к) $<>$ | 10) склеивание двух текстовых аргументов |
| л) $<=$ | 11) сравнение аргументов (если аргументы равны) |

9.5. Для осуществления более сложных действий над аргументами в электронных таблицах используются функции. Укажите, какие из приведенных утверждений верны при записи функций (в электронной таблице). Буквы, соответствующие правильным утверждениям, обведите, остальные — зачеркните:

- а) функции используются в формулах (то есть в выражениях, начинающихся со знака $=$ (равно));
- б) каждая функция имеет свое уникальное имя;
- в) имя функции должно быть записано обязательно русскими буквами;
- г) после указания имени функции нужно обязательно написать аргументы в круглых скобках;

- д) даже если функция не имеет аргументов, все равно нужно написать пустые круглые скобки;
- е) аргументы функции отделяются друг от друга символом «;» (точка с запятой);
- ж) для указания в качестве аргумента целой области соседних ячеек используется *диапазон* ячеек;
- з) прямоугольный диапазон ячеек записывается указанием адреса левой верхней и правой нижней ячейки прямоугольного диапазона через символ «:» (двоеточие). Например, А1:В3;
- и) для указания в качестве диапазона целиком строки, столбца или нескольких используется только один (нужный) индекс. Например: А:А (весь столбец А); В:Д (столбцы от В до Д); 6:6 (вся строка 6); 4:12 (строки начиная с 4-й по 12-ю).

9.6. В списке наиболее часто используемых функций установите соответствие между именем функции и возвращаемым ею значением. Впишите рядом с буквой функции число, соответствующее смыслу этой функции:

- | | |
|-----------|-------------------------------------------------------------------------------|
| а) СУММ | 1) наибольшее значение |
| б) СРЗНАЧ | 2) если указанное условие ИСТИНА, то второй аргумент, иначе — третий аргумент |
| в) МАКС | 3) наименьшее значение |
| г) МИН | 4) сумма |
| д) ЕСЛИ | 5) среднее арифметическое (сумма, деленная на количество) |

9.7. Во всех ячейках таблицы записано число 1. Вычислите результат функции СУММ для каждого списка аргументов:

- а) СУММ(А2) = _____
- б) СУММ(А2:А4) = _____
- в) СУММ(С3:F3) = _____
- г) СУММ(В2:Д5) = _____
- д) СУММ(А3:А7;В4:С5) = _____
- е) СУММ(А1;В3;А4;В8) = _____
- ж) СУММ(А5;В3;Е6;F12) = _____
- з) СУММ(3:3) = _____

9.8. Для данной электронной таблицы вычислите результат функций:

	А	В
1	3	5
2	2	9
3	5	7
4	0	3

- а) МИН(А1:В4) _____
- б) МАКС(А3:В4) _____
- в) СУММ(А1:А4) _____
- г) СРЗНАЧ(А1:А4) _____
- д) СРЗНАЧ(В1:В4) _____

9.9. В электронной таблице значение формулы =СУММ(B1:B2) равно 5. Чему равно значение ячейки B3, если значение формулы =СРЗНАЧ(B1:B3) равно 3?

- 1) 8 2) 2 3) 3 4) 4

Рекомендация. Нарисуйте на черновике (или представьте) упоминаемые в задаче ячейки. Помните, что символ «:» (двоеточие) между адресами ячеек — это не перечисление, а указание на диапазон. Помните, что СРЗНАЧ — это не «сложить и поделить пополам», а сумма указанных ячеек, деленная на их количество. А количество в данном случае вовсе не две. Догадайтесь, что если мы знаем количество и среднее значение, то можно отсюда вычислить сумму. Осталось догадаться, что если известна сумма N чисел и сумма тех же чисел, но без одного из них, то вычитанием легко вычислить оставшееся число.

9.10. В электронной таблице значение формулы =СРЗНАЧ(A1:C1) равно 5. Чему равно значение ячейки D1, если значение формулы =СУММ(A1:D1) равно 7?

- 1) 2 2) -8 3) 8 4) -3

9.11. В электронной таблице значение формулы =СРЗНАЧ(B1:D1) равно 4. Чему равно значение ячейки A1, если значение формулы =СУММ(A1:D1) равно 9?

- 1) -3 2) 5 3) 1 4) 3

9.12. В электронной таблице значение формулы =СРЗНАЧ(A1:B4) равно 3. Чему равно значение ячейки A4, если значение формулы =СУММ(A1:B3) равно 30, а в ячейке B4 лежит 5?

Ответ: _____.

Напоминание. Обычная запись адреса ячейки в формуле (например, C5) соответствует относительной адресации ячейки. Это значит, что при копировании такой формулы в другую ячейку таблицы произойдет автоматическая индексация. То есть обе части адреса ячейки (номер строки и номер столбца, они называются *индексами*) изменятся на такое значение, на сколько по вертикали и по горизонтали ячейка была скопирована относительно своего исходного положения. Например, формула «=C5» находится в ячейке A4. Эта формула копируется в ячейку B2 — на 2 строки выше и на один столбец правее. То есть новый адрес формулы уменьшается на 2 по вертикали (номер строки) и увеличивается на один по горизонтали (номер столбца). Значит, относительные адреса ячеек, используемых в формуле, изменяются на эти же величины: номер строки уменьшается на 2, номер столбца увеличивается на 1. Адрес C5 превратится в адрес D3 (C + 1, 5 - 2).

9.13. Дан фрагмент электронной таблицы:

	A	B	C
1	1	3	=A1+B1+2
2	2	4	

Чему станет равным значение ячейки C2, если в нее скопировать формулу из ячейки C1?

Ответ: _____.

Напоминание. Если при копировании формулы не требуется автоматическая индексация, используют абсолютную адресацию ячеек. Для этого перед той частью адреса (перед тем индексом), которая не должна измениться при копировании, нужно поставить знак доллара (\$). Если знак абсолютной адресации стоит перед обеими частями адреса (обеими индексами, например \$C\$5), этот адрес не изменится при копировании в любое место. Если знак «\$» только перед одной частью адреса — будет изменяться та часть адреса (тот индекс), перед которым знак доллара не стоит. Например, адрес \$B4 не будет меняться при копировании влево-вправо, но будет меняться (как обычно) при копировании вверх-вниз.

9.14. Дан фрагмент электронной таблицы:

	A	B	C
1	10	20	=A1+B\$1+2
2	30	40	

Чему станет равно значение ячейки C2, если в нее скопировать формулу из ячейки C1?

Ответ: _____.

9.15. При работе с электронной таблицей в ячейку C3 записана формула: =\$B3+C\$2. Какой вид приобретет формула, после того как ячейку C3 скопируют в ячейку D2?

Ответ: _____.

9.16. При работе с электронной таблицей в ячейку E5 записана формула: =\$C3-B\$3. Какой вид приобретет формула, после того как ячейку E5 скопируют в ячейку D3?

Ответ: _____.

9.17. При работе с электронной таблицей в ячейку B4 записана формула: =\$A\$4+C3. Какой вид приобретет формула, после того как ячейку B4 скопируют в ячейку C2?

Ответ: _____.

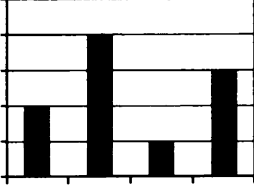
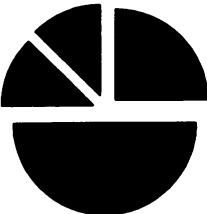
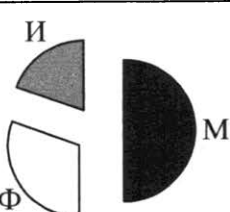
9.18. В ячейке B1 записана формула: =2 * \$A1. Какой вид приобретет формула, после того как ячейку B1 скопируют в ячейку C2?

Ответ: _____.

Напоминание. Для наглядного представления табличной информации часто используют диаграммы. В зависимости от необходимого визуального эффекта выбирают тип диаграммы. В основном это *Линейчатая* и *Гистограмма* (горизонтальные или вертикальные прямоугольники), *Кольцевая* и *Круговая*, а также *График* и *Точечная*.

Основное, что вы должны уметь делать, — понимать, что диаграммы строятся на основе данных, находящихся в таблице и что Кольцевая и Круговая диаграммы отображают соотношение этих данных к их общей сумме. А в остальных диаграммах отображаются абсолютные значения.

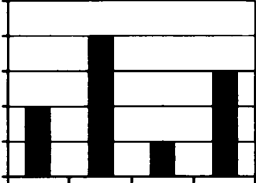
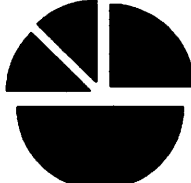
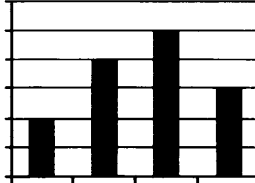

Примеры.

Диаграмма	Смысл диаграммы
	<p>Если принять, что единица измерения вдоль вертикальной оси равна одному, то диаграмма построена по данным «2, 4, 1, 3».</p>
	<p>Данные, по которым построена диаграмма, соотносятся между собой как 1:1:2:4.</p>
	<p>Значение М составляет половину от общей суммы М+И+Ф. Значение Ф чуть больше четверти от общей суммы, значение И — меньше четверти.</p>

9.19. Дан фрагмент электронной таблицы:

	A	B
1	=B1+1	1
2	=A1+2	2
3	=B2-1	
4	=A3	

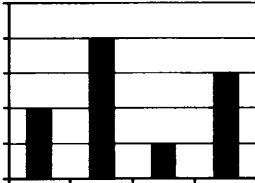
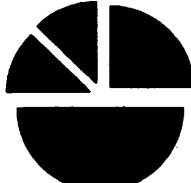
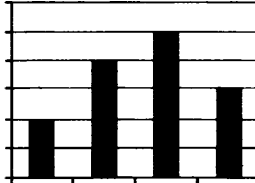

После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A1:A4. Укажите получившуюся диаграмму.

1)  2)  3)  4) 

9.20. Дан фрагмент электронной таблицы:

	A	B	C	D
1	3		5	
2	=A1/3	=(A1+C1+1)/3	=C1-2	=(B2+C2)/6

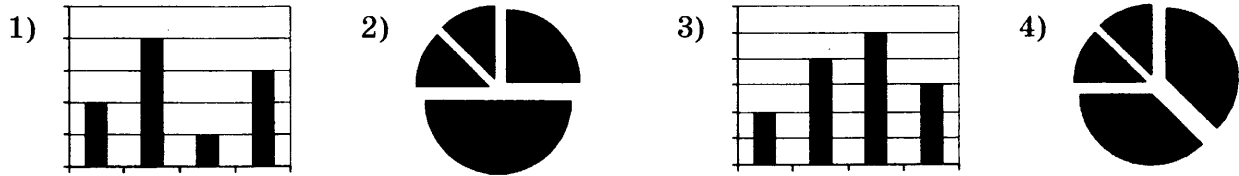
После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A2:D2. Укажите получившуюся диаграмму.

1)  2)  3)  4) 

9.21. Дан фрагмент электронной таблицы:

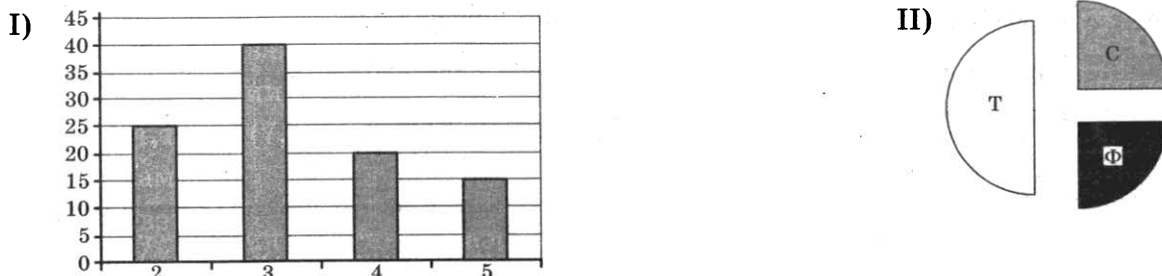
	A	B	C	D
1	3		5	
2	$=(A1+C1)/4$	$=C1-1$	$=A2/2$	$=B2-C2$

После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A2:D2. Укажите получившуюся диаграмму.



9.22. В цехе трудятся рабочие трех специальностей — токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд не меньше второго и не больше пятого. На диаграмме I отражено количество рабочих с различными разрядами, а на диаграмме II — распределение рабочих по специальностям.

Каждый рабочий имеет только одну специальность и один разряд.



Имеются четыре утверждения:

- А) все рабочие третьего разряда могут быть токарями;
- Б) все рабочие третьего разряда могут быть фрезеровщиками;
- В) все слесари могут быть пятого разряда;
- Г) все токари могут быть четвертого разряда.

Какое из этих утверждений следует из анализа обеих диаграмм?

- 1) А 2) Б 3) В 4) Г

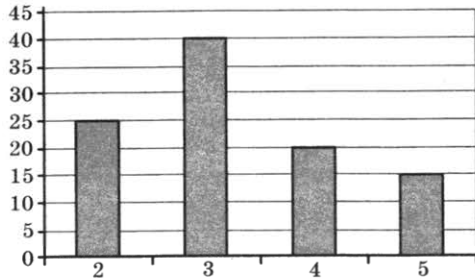
Рекомендации. По гистограмме (I диаграмма) посчитайте, сколько всего рабочих в цехе (второго разряда — 25, третьего — 40, четвертого — 20, пятого — 15; всего $25 + 40 + 20 + 15 = 100$). Пользуясь соотношением, полученным из круговой диаграммы (II диаграмма), вычислите количество токарей, слесарей и фрезеровщиков (токарей половина, то есть $100 / 2 = 50$; фрезеровщиков и слесарей по четверти, то есть по $100 / 4 = 25$). Проверьте получившиеся соотношения:

- А) рабочих 3-го разряда 40, а токарей — 50. Могут ли быть все 40 рабочих 3-го разряда среди 50 токарей? Да, могут;
- Б) рабочих 3-го разряда 40, а фрезеровщиков — 25. Могут ли быть все 40 рабочих 3-го разряда 25 фрезеровщиками? Нет, не могут;
- В) слесарей всего 25, а рабочих 5-го разряда — 15. Могут ли быть все 25 слесарей 15 рабочими 5-го разряда? Нет, не могут;
- Г) токарей всего 50, а рабочих 4-го разряда — 20. Могут ли быть все 50 токарей 20 рабочими 4-го разряда? Нет, не могут.

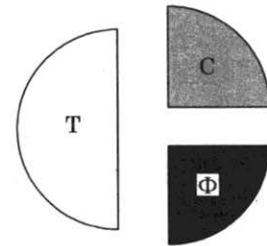
9.23. В цехе трудятся рабочие трех специальностей — токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд не меньше второго и не больше пятого. На диаграмме I отражено количество рабочих с различными разрядами, а на диаграмме II — распределение рабочих по специальностям.

Каждый рабочий имеет только одну специальность и один разряд.

I)



II)



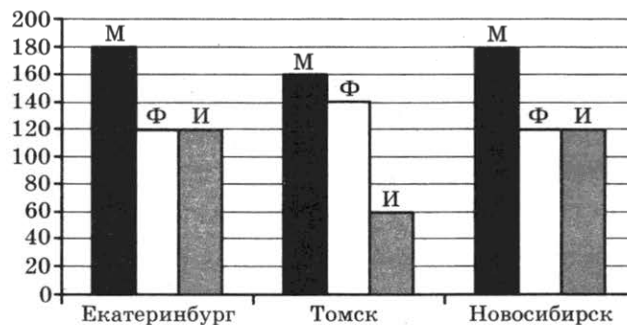
Имеются четыре утверждения:

- все рабочие второго разряда могут быть слесарями;
- все фрезеровщики могут быть четвертого разряда;
- среди рабочих третьего разряда может не быть ни одного токаря;
- у всех токарей разряд может быть не меньше четвертого.

Какие из этих утверждений следуют из анализа обеих диаграмм?

Обведите буквы, соответствующие этим утверждениям.

9.24. На диаграмме показано количество призеров олимпиады по информатике (И), математике (М), физике (Ф) в трех городах России.



Какая из диаграмм правильно отражает соотношение общего числа призеров по каждому предмету для всех городов вместе?

-
-
-
-

Пример решения

По первой диаграмме получим общее количество призеров по каждому предмету.

В Екатеринбурге призеров по математике 180, в Томске — 160, в Новосибирске — 180.

Итого призеров по математике $180 + 180 + 160 = 520$.

Аналогично получим, что призеров по физике всего $120 + 140 + 120 = 380$, а призеров по информатике — $120 + 60 + 120 = 300$.

Сложим полученные три суммы и получим, что общее число призеров по всем трем предметам равно $520 + 380 + 300 = 1200$.

Найдем отношение числа призеров по каждому предмету к общему числу призеров.

По математике — $520/1200$ — чуть меньше половины.

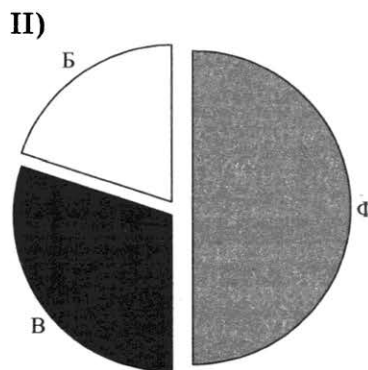
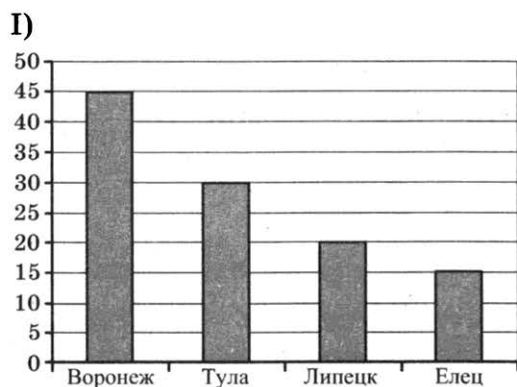
По физике — $380/1200$ — примерно треть.

По информатике — $300/1200$ — ровно четверть.

На приведенных диаграммах только первая отображает такое соотношение призеров.

Рекомендация. По первой диаграмме получите общее количество призеров по каждому предмету (сложите). Сложите получившиеся числа и посмотрите, как общее число призеров по каждому предмету соотносится с их общим количеством. Прикиньте, как каждое из получившихся отношений соотносится с половиной и с четвертью. Найдите получившееся соотношение среди приведенных диаграмм.

9.25. В спортивный пионерский лагерь приехали дети из четырех городов, занимающихся тремя видами спорта: футбол, волейбол и баскетбол. На диаграмме I показано, сколько детей приехало из какого города. На диаграмме II показано распределение детей по видам спорта.



Имеются четыре утверждения:

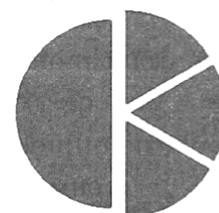
- все футболисты могут быть из Воронежа;
- среди детей из Тулы может не быть ни одного футболиста;
- среди волейболистов может не быть ни одного ребенка из Липецка;
- среди детей из Тулы больше футболистов, чем баскетболистов.

Какие из этих утверждений следуют из анализа обеих диаграмм?

Обведите буквы, соответствующие этим утверждениям.

9.26. Дан фрагмент электронной таблицы.

	A	B	C	D
1	6			
2	=D2-B1	=B1/2	=A1-B2	=A1+B1/2



Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?

9.30. При работе с электронной таблицей решили составить таблицу произведений четных и нечетных цифр. Для этого в ячейку В2 записали формулу, которую затем скопировали в остальные ячейки диапазона В2:Е6. Укажите формулу, записанную в ячейке В2.

	A	B	C	D	E
1		2	4	6	8
2	1	2	4	6	8
3	3	6	12	18	24
4	5	10	20	30	40
5	7	14	28	42	56
6	9	18	36	54	72

1) =A2*B1

2) =\$A2*\$B1

3) =\$A2*B\$1

4) =A\$2*\$B1

10. БАЗЫ ДАННЫХ

Основные понятия

Одной из самых удобных информационных моделей является база данных.

База данных — информационная модель, хранящая данные о группе объектов с одинаковым набором свойств. Как правило, база данных хранит информацию о нескольких группах объектов, связанных между собой.

В школьном курсе информатики изучают только **реляционные базы данных**. В них информация хранится в виде таблиц, связанных между собой отношениями подчинения.

Каждая строка таблицы описывает один объект. Этот объект может иметь различные свойства, информация о которых хранится в столбцах таблицы.

Данные в строках, таким образом, могут быть разных типов. Одна строка называется **записью**. Данные в каждом столбце описывают одно свойство объектов. Таким образом, данные в одном столбце имеют одинаковый тип. Столбец таблицы называется **полем**.

Для указания на конкретный объект (запись) таблицы принято вводить специальный, уникальный набор полей (или одно поле). Такой набор называется **ключом**, а поле — **ключевым**. Самый надежный и удобный способ — в каждой таблице хранить поле с уникальным кодом строки (записи).

Для связи между таблицами в подчиненной таблице используются специальные поля, в которых хранится информация о связанных записях главной таблицы. Обычно — ключ.

Для выбора из базы данных объектов с определенными свойствами используются операции сравнения, а также арифметические и логические операции.

Так как в школьном курсе информатики изучаются только реляционные базы данных, все сказанное ниже относится именно к реляционным базам данных.

Практическая часть

10.1. Укажите одно или несколько правильных утверждений. Буквы, соответствующие правильным утверждениям, обведите, остальные — зачеркните.

Для баз данных справедливо:

- а) информация хранится в виде таблиц, связанных отношениями подчинения;
- б) строки таблицы хранят информацию об объектах и называются записями;
- в) столбцы таблицы хранят информацию о свойствах объектов;
- г) информация в столбце всегда одного типа данных;
- д) столбец таблицы называется полем.

Напоминание. Для осуществления каких-либо действий над записями баз данных необходимо указывать, над какими именно записями нужно осуществить эти действия. Для отбора нужных записей используют условия. В условиях указываются имена полей, которые сравниваются друг с другом или с константами при помощи операций соотношения. Эти операции те же, что и в электронных таблицах.

10.2. Сопоставьте каждому сравнению (слева) условие, которое это сравнение проверяет (справа). Впишите рядом с буквой сравнения число, соответствующее смыслу этого сравнения:

- | | |
|-----------|--------------------------------------------------|
| а) $>$ | 1) если первый аргумент меньше или равен второму |
| б) $=$ | 2) если первый аргумент меньше второго |
| в) \geq | 3) если первый аргумент больше или равен второму |
| г) $<>$ | 4) если первый аргумент больше второго |
| д) \leq | 5) если аргументы не равны |
| е) $<$ | 6) если аргументы равны |

10.3. В таблице представлены результаты тестирования:

Фамилия	Математика	Физика
Иванов	76	54
Петров	38	74
Сидоров	40	43
Орлов	67	67
Пастухов	38	95
Николаев	23	37

Сколько записей в вышеприведенной таблице удовлетворяют условию «Математика > 38 »?

Ответ: _____.

10.4. Сколько записей в вышеприведенной таблице удовлетворяют условию «Физика ≤ 54 »?

Ответ: _____.

10.5. Сколько записей в вышеприведенной таблице удовлетворяют условию «Физика \geq Математика»?

Ответ: _____.

10.6. Сколько записей в вышеприведенной таблице удовлетворяют условию «Физика $<>$ Математика»?

Ответ: _____.

10.7. Сколько записей в вышеприведенной таблице удовлетворяют условию «Математика $<> 38$ »?

Ответ: _____.

Напоминание. Если требуется проверить сразу несколько условий, возникает необходимость объединить эти условия. Для этого, как вы помните из алгебры логики, используются логические операции.

10.8. Сопоставьте каждой логической операции (слева) ее смысл (справа). Впишите рядом с буквой операции число, соответствующее смыслу этой операции:

- | | |
|--------|--------------------------------------------------------------------------------------------------|
| а) И | 1) результат меняется на противоположный.
Если операнд — ИСТИНА, результат — ЛОЖЬ. И наоборот |
| б) ИЛИ | 2) результат — ИСТИНА, если оба операнда — ИСТИНА |
| в) НЕ | 3) результат — ИСТИНА, если хотя бы один операнд — ИСТИНА |

Напоминание. Если в выражении присутствует несколько логических операций, они выполняются в соответствии со своими приоритетами.

10.9. Укажите порядок выполнения операций в соответствии с их приоритетом (от высшего к низшему). В ответе укажите буквы, соответствующие операциям, в указанном порядке.

- а) ИЛИ б) выражение в скобках в) И г) НЕ

Ответ: _____.

10.10. Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяют условию «Пол='ж' ИЛИ Химия>Биология»?

- 1) 5 2) 2 3) 3 4) 4

Рекомендация. Выберите записи, которые удовлетворяют первому условию (Пол='ж'). Отметьте их (например, «галочками»). Затем выберите записи, удовлетворяющие второму условию (Химия>Биология). Также отметьте их «галочками». Так как для их объединения используется логическая операция ИЛИ, ей удовлетворяют те записи (строки), в которых мы поставили хотя бы одну «галочку».

10.11. Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяют условию

«НЕ(Пол='ж' И Математика<Биология)»?

- 1) 5 2) 2 3) 3 4) 4

Рекомендация. Такие задания (с операцией НЕ перед всем выражением) можно решать двумя способами. Либо выберите записи, которые удовлетворяют выражению. Тогда все остальные будут ему НЕ удовлетворять, то есть именно то, что нужно найти.

Либо примените закон Де Моргана. Но в этом случае нужно быть очень осторожным и не забыть обратить все условия. В данном случае:

НЕ(Пол='ж' И Математика<Биология) =

НЕ(Пол='ж') ИЛИ НЕ(Математика<Биология) =

Пол<>'ж' ИЛИ Математика>=Биология

10.12. Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяют условию

«Пол='ж' И Математика>Химия»?

1) 5

2) 2

3) 3

4) 4

10.13. Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяют условию

«Пол='ж' И Математика>Русский язык ИЛИ Пол='м' И Информатика>Биология»?

1) 5

2) 2

3) 3

4) 4

Рекомендация. Не забудьте, что приоритет операции «Логическое И» выше, чем у операции «Логическое ИЛИ».

10.14. Из правил соревнования по тяжелой атлетике:

«Тяжелая атлетика — это прямое соревнование, когда каждый атлет имеет три попытки в рывке и три попытки в толчке. Самый тяжелый вес поднятой штанги в каждом упражнении суммируется в общем зачете. Если спортсмен потерпел неудачу во всех трех попытках в рывке, он может продолжить соревнование в толчке, но уже не сможет занять какое-либо место по сумме двух упражнений.

Если два спортсмена заканчивают состязание с одинаковым итоговым результатом, высшее место присуждается спортсмену с меньшим весом. Если же вес спортсменов одинаков, преимущество отдается тому, кто первым поднял победный вес».

Таблица результатов соревнований по тяжелой атлетике:

Фамилия, И.О.	Вес спортсмена	Взято в рывке	Рывок с попытки	Взято в толчке	Толчок с попытки
Айвазян Г.С.	77,1	150,0	3	200,0	2
Викторов М.П.	79,1	147,5	1	202,5	1
Гордезиани Б.Ш.	78,2	147,5	2	200,0	1
Михальчук М.С.	78,2	147,5	2	202,5	3
Пай С.В.	79,5	150,0	1	200,0	1
Шапсугов М.Х.	77,1	147,5	1	200,0	1

Кто победил в общем зачете (сумме двух упражнений)?

- | | |
|------------------|-------------------|
| 1) Айвазян Г.С. | 3) Михальчук М.С. |
| 2) Викторов М.П. | 4) Пай С.В. |

10.15. В таблице собраны сведения о членах школьной волейбольной команды:

Фамилия	Вес	Рост	Пол
Иванов	87	180	М
Петрова	55	170	Ж
Сидоров	67	155	М
Пупкина	78	160	Ж

Сколько записей удовлетворяют условию:

девочки выше 165 см И легче 60 кг ИЛИ учащиеся тяжелее 75 кг?

Ответ: _____.

10.16. На городской олимпиаде по программированию предлагались задачи трех типов: А, В и С. По итогам олимпиады была составлена таблица, в колонках которой указано, сколько задач каждого типа решил участник. Вот начало таблицы:

Фамилия	А	В	С
Иванов	3	2	1

За правильное решение задачи типа А участнику начислялся 1 балл, за решение задачи типа В — 2 балла и за решение задачи типа С — 3 балла. Победитель определялся по сумме баллов, которая у всех участников оказалась разная. Какой запрос достаточно выполнить для определения победителя олимпиады из следующих:

- 1) отсортировать таблицу по возрастанию значения поля С и взять первую строку
- 2) отсортировать таблицу по убыванию значения поля С и взять первую строку
- 3) отсортировать таблицу по убыванию значения выражения $A + 2B + 3C$ и взять первую строку
- 4) отсортировать таблицу по возрастанию значения выражения $A + 2B + 3C$ и взять первую строку

Рекомендация. Проанализируйте принцип начисления баллов и выведите формулу суммы баллов. Если победитель выбирается из первой строки запроса, проанализируйте, какое количество баллов он должен набрать — наибольшее или наименьшее? Как в этом случае должна быть отсортирована таблица?

10.17. На городской олимпиаде по программированию предлагались задачи трех типов: А, В и С. По итогам олимпиады была составлена таблица, в колонках которой указано, сколько задач каждого типа решил участник. Вот начало таблицы:

Фамилия	А	В	С
Иванов	4	1	3

За правильное решение задачи типа А участнику начислялось 3 балла, за решение задачи типа В — 2 балла и за решение задачи типа С — 1 балл. Победитель определялся по сумме баллов, которая у всех участников оказалась разная. Какой запрос достаточно выполнить для определения победителя олимпиады из следующих:

- 1) отсортировать таблицу по возрастанию значения поля С и взять первую строку
- 2) отсортировать таблицу по убыванию значения поля С и взять первую строку
- 3) отсортировать таблицу по возрастанию значения выражения $3A + 2B + C$ и взять первую строку
- 4) отсортировать таблицу по убыванию значения выражения $3A + 2B + C$ и взять первую строку

10.18. Ниже приведены фрагменты таблиц базы данных канцелярского магазина:

Вид изделия	Код изделия
Авторучка	001
Фломастер	002
Карандаш	003

Цвет изделия	Код цвета
синий	01
зеленый	02
красный	03

Размер изделия	Код размера
большой	0001
маленький	0002

Артикул	Изделие	Размер	Цвет	Цена
8457	002	0002	01	5
2537	003	0001	01	9
4758	002	0002	02	5
5748	003	0001	02	8
3647	001	0001	02	9
1948	003	0002	01	6
3937	001	0001	01	8
3524	002	0001	03	8
1362	002	0002	03	6

10.21. Ниже приведены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребенке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведенных данных фамилию и инициалы деда Глянко Е.Е.:

Таблица 1

ID	Фамилия И.О.	Пол
14	Шичко К.Д.	Ж
17	Карно Г.Е.	М
21	Басик О.Б.	Ж
26	Гилага И.Е.	Ж
34	Красик Д.О.	М
48	Арис И.Г.	М
53	Глянко Е.Е.	М
57	Орба А.А.	М
71	Варус Е.А.	М
78	Гудько Г.Е.	Ж
82	Асава С.К.	Ж

Таблица 2

ID_Родителя	ID_Ребенка
71	26
53	17
14	53
34	14
14	26
17	48
71	53
57	78
78	14
26	82
21	17

- 1) Гудько Г.Е. 2) Арис И.Г. 3) Красик Д.О. 4) Карно Г.Е.

10.22. Те же таблицы, что в задаче 10.21. Определите на основании приведенных данных фамилию и инициалы племянницы Глянко Е.Е. (племянница — дочь брата или сестры):

- 1) Шичко К.Д. 2) Асава С.К. 3) Басик О.Б. 4) Гилага И.Е.

10.23. Те же таблицы, что в задаче 10.21. Определите на основании приведенных данных фамилию и инициалы тети Карно Г.Е. (тетя — сестра отца или матери):

- 1) Басик О.Б. 2) Шичко К.Д. 3) Асава С.К. 4) Гилага И.Е.

11. СЕТЕВЫЕ ТЕХНОЛОГИИ

Основные понятия

Адресация в Интернете. Для обмена информацией между компьютерами необходимы:

- канал связи, по которому идет передача данных;
- аппаратные средства, которыми компьютер связывается с каналом связи (модем, сетевая карта);
- программные средства, которыми при этом пользуется компьютер или человек (браузер, почтовая программа, менеджер обмена сообщениями);
- протокол (соглашение, согласно которому происходит процедура обмена).

Для соединения через Интернет каждый компьютер должен обладать протоколом TCP/IP. Согласно этому протоколу, в частности, каждый компьютер должен иметь уникальный адрес, состоящий из четырех чисел (от 0 до 255 каждое), разделенных точками (например, 172.16.1.95). Это называется **IP-адрес**. Так как каждое число IP-адреса лежит в диапазоне 0–255, нетрудно подсчитать, что для его хранения требуется 8 бит, то есть 1 байт.

Также нетрудно подсчитать, что общее количество IP-адресов не может быть больше чем 256^4 . То есть около 4 миллиардов. Это, однако, не означает, что к Интернету не может быть подключено больше 4 миллиардов компьютеров. Специальные технологии позволяют целой группе компьютеров использовать один общий IP-адрес для выхода в Интернет.

Некоторые компьютеры в сети предоставляют свои ресурсы в общее пользование. Они называются **серверами**. Как правило, это серверы с информацией. Так как множество людей часто обращаются к серверам, неудобно помнить для каждого нужного сервера его IP-адрес. Для таких серверов используются буквенные названия, которые называются **доменным адресом**. Например, www.mail.ru. Когда нужно обратиться к серверу по доменному адресу, компьютер сначала посылает запрос в специальную службу (она называется DNS), которая по доменному адресу выдает IP-адрес этого сервера.

Доменные адреса обычно выбирают по территориальному принципу или по принципу принадлежности сервера к организации какого-либо типа.

Пример территориального принципа выбора доменного адреса: sch239.spb.ru (Россия, Санкт-Петербург, школа 239).

По принципу принадлежности организации к какому-либо типу обычно действуют организации из США (родины Интернета). Примеры: whitehouse.gov, pentagon.mil, greenpeace.org, mti.edu.

Для указания местоположения какого-либо ресурса (файла) в сети Интернет используют URL (универсальный указатель местоположения ресурса). Он состоит из трех частей: имени протокола, по которому будет происходить передача файла, затем через символы «://» адрес компьютера, на котором этот файл хранится (IP-адрес или доменный адрес), затем через символ «/» имя файла на этом компьютере (иногда с указанием пути к файлу). Пример: http://masha.ru/risunok.jpg.

Для поиска информации в Интернете используют **поисковые системы** — специальные программно-аппаратные комплексы (базы данных), которые хранят информацию о множестве страниц Интернета и которые предоставляют возможность поиска среди этой информации.

Для поиска нужно указать ключевые слова, которые должны содержаться в искомом документе. Чтобы указать, должны ли эти слова присутствовать в документе одновременно

или же достаточно одного из них, используют логические операции (И и ИЛИ). Операцию И обозначают значком «&» (амперсанд), операцию ИЛИ обозначают значком «|» (вертикальная черта). Приоритет операции И выше, чем у операции ИЛИ.

Для изменения порядка выполнения операций используют скобки.

Если несколько ключевых слов объединены операцией И, количество найденных страниц будет меньше, чем если эти слова искать по отдельности.

Если несколько ключевых слов объединены операцией ИЛИ, количество найденных страниц будет, наоборот, больше.

Практическая часть

11.1. Укажите одно или несколько правильных утверждений. Буквы, соответствующие правильным утверждениям, обведите, остальные — зачеркните.

Для компьютерной сети Интернет справедливо:

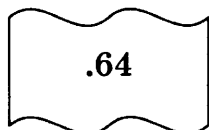
- а) каждый компьютер, подключенный к сети, должен иметь собственный адрес — IP-адрес
- б) IP-адрес состоит из четырех чисел, разделенных точками
- в) для хранения каждого из чисел IP-адреса выделяют по 1 байту
- г) каждое из чисел IP-адреса может быть от 0 до 255
- д) некоторые числа IP-адреса можно заменять на слова (например, www.mail.ru)
- е) некоторые комбинации чисел IP-адреса неприемлемы ввиду договоренностей об IP-адресах
- ж) наибольшее значение (255) для чисел в IP-адресе происходит от размера ячейки памяти, выделяемой для IP-адреса
- з) общее количество IP-адресов, которое может существовать, составляет 256^4 (примерно 4 миллиарда)
- и) количество компьютеров в сети не может превышать четырех миллиардов ввиду ограничения на количество возможных IP-адресов

11.2. Сколько точек в IP-адресе?

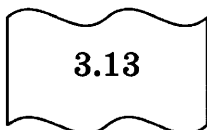
Ответ: _____.

11.3. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

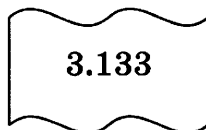
В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



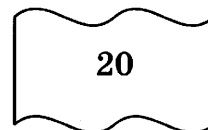
А



Б



В



Г

Ответ: _____.

Рекомендация. Честно говоря, мы заметили, что проще всего решать такую задачу, действительно написав на четырех клочках бумаги куски IP-адреса и перекладывая их. Правда, мы не уверены, что если вы начнете заниматься этим на экзамене, вам не сделают замечание.

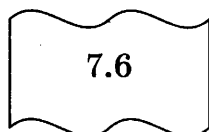
Основная идея решения этой задачи — в недопустимости чисел IP-адреса, которые превышают 255. В данном случае задача решается совсем просто, ввиду быстрого установления двух обрывков на свои места. Обрывок А не может быть первым (из-за точки). Кроме того, так как число в нем больше 25, то справа к нему нельзя дописать ни одной цифры, иначе получатся числа больше 640. Значит, 64 — одно из чисел IP-адреса, причем не первое. А так как больше обрывков, начинающихся с точки, нет, значит, обрывок А стоит последним.

К обрывку В не может быть приставлена справа ни одна цифра, так как число 133 уже трехзначное. Значит, обрывок В должен быть или последним, или после него должна стоять точка. Последнее место уже занято. Значит, обрывок В предпоследний.

Оставшиеся 2 варианта нетрудно просто проверить.

11.4. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

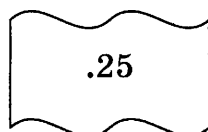
В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



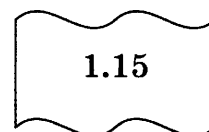
А



Б



В

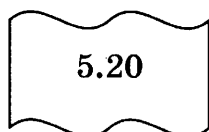


Г

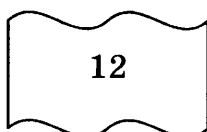
Ответ: _____.

11.5. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

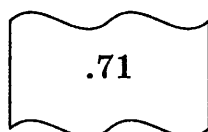
В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



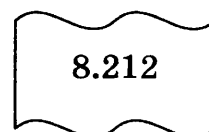
А



Б



В



Г

Ответ: _____.

11.6. Укажите одно или несколько правильных утверждений. Буквы, соответствующие правильным утверждениям, обведите, остальные — зачеркните.

- Для указания местоположения ресурса в сети Интернет используется специальный адрес, называемый URL
- URL — Uniform Resource Locator

- в) URL состоит из трех частей — протокол, символы «://», адрес компьютера в сети, символ «/», местоположение файла на этом компьютере
- г) Пример URL: «http://www.prazdnik.ru/files/holyday.jpg»
- д) Протокол — договоренность о способе обмена информацией
- е) Примеры наиболее часто используемых протоколов: http, ftp
- ж) Адрес компьютера в URL может быть доменным или ip-адресом
- з) Допустимым считается, например, такой URL: «ftp://10.20.4.58/index.html»
- и) Местоположение файла в URL может включать в себя последовательность имен папок от «стартового положения» файлов на сервере, разделенных символами «/»
- к) URL всегда заканчивается именем файла

11.7. Доступ к файлу htm.net, находящемуся на сервере com.edu, осуществляется по протоколу ftp. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	/
Б	com
В	.edu
Г	://
Д	.net
Е	htm
Ж	ftp

Ответ: _____

Рекомендация. Чтобы не ошибиться в символах и буквах, сначала аккуратно запишите на черновике получающийся URL. Не забудьте записать между протоколом и именем сервера символы «://», а между именем сервера и именем файла — символ «/». Не старайтесь использовать привычные имена серверов и имена файлов. Не старайтесь расставлять их в том порядке, как вы привыкли ими пользоваться. Просто «тупо» запишите последовательность, которая указана в задаче: протокол, «://», имя сервера, «/», имя файла. Потом подпишите под каждой из получившихся частей URL свою букву по таблице. Не забудьте проверить, что у вас получились все 7 букв и что они не повторяются.

11.8. Доступ к файлу www.txt, находящемуся на сервере http.net, осуществляется по протоколу ftp. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	http
Б	.txt
В	.net
Г	ftp
Д	/
Е	www
Ж	://

Ответ: _____

11.9. Доступ к файлу `txt.com`, находящемуся на сервере `www.ftp`, осуществляется по протоколу `http`. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	.ftp
Б	txt
В	://
Г	www
Д	http
Е	/
Ж	.com

Ответ: _____

11.10. Доступ к файлу `com.ftp`, находящемуся на сервере `www.edu`, осуществляется по протоколу `http`. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	.ftp
Б	/
В	.edu
Г	://
Д	www
Е	com
Ж	http

Ответ: _____

Напоминание. Поиск информации в Интернете.

При пользовании поисковыми системами запрос, который делает пользователь, обрабатывается базой данных, хранящейся на поисковом сервере. Ключевые слова, которые вводит пользователь в тексте запроса, ищутся по отдельности в базе данных, после чего результаты каждого поиска соединяются в зависимости от логических операций, которыми объединены ключевые слова. Если используется логическая операция «И», то в результат войдут только те страницы, которые содержат одновременно оба ключевых слова (мы говорим «оба», так как логические операции «И» и «ИЛИ» всегда имеют ровно два операнда). Значит, в результате поиска количество страниц, которое найдет поисковый сервер, будет меньше, чем в запросах по каждому ключевому слову в отдельности.

Если используется логическая операция «ИЛИ», то в результат войдут как страницы, которые содержат только одно из ключевых слов, так и страницы, которые содержат только второе ключевое слово. Также результат будет содержать страницы, которые содержат сразу оба ключевых слова. Значит, в результате поиска количество страниц, которое найдет поисковый сервер, будет больше, чем по каждому ключевому слову в отдельности.

Не забудьте, что приоритеты логических операций здесь действуют как обычно. Приоритет операции «И» выше, чем у операции «ИЛИ» (см. главу 4 «Основы логики»).

Замечание. Мы, конечно, слегка грешим против истины, для упрощения понимания. В действительности правильнее говорить, что результат запроса с логической операцией «И» не больше результата запроса с каждым ключевым словом в отдельности. Так же как и результат запроса с логической операцией «ИЛИ» не меньше запросов с этими словами по отдельности.

Эта оговорка работает, когда, например, в базе данных существуют только страницы, в которых оба ключевых слова встречаются одновременно, а по отдельности эти ключевые слова не встречаются. Но мы такой случай для общности и простоты задачи не рассматриваем. Вам мы советуем при решении такого типа задач тоже считать, что это не наш случай.

11.11. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	принтеры продажа
2	принтеры
3	принтеры & продажа

Ответ: _____.

11.12. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	коньки лыжи
2	коньки & лыжи & санки
3	коньки & лыжи
4	лыжи

Ответ: _____.

11.13. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	принтеры & сканеры & продажа
2	принтеры & продажа
3	принтеры продажа
4	принтеры сканеры продажа

Ответ: _____.

Рекомендация. Учтите, что условие «логическое И» приводит к уменьшению количества найденных страниц. Соответственно, применение нескольких «логических И» еще более уменьшает число найденных страниц.

В отличие от «И», «логическое ИЛИ» приводит к увеличению числа найденных страниц. Соответственно, применение нескольких «логических ИЛИ» еще более увеличивает количество найденных страниц.

Постарайтесь быть внимательным и обратить внимание на порядок возрастания/убывания. При возрастании нужно сначала указать самое строгое условие (наименьшее число найденных страниц), а в конце — самое «широкое» (больше всего «ИЛИ»). При убывании — наоборот, в обратном порядке.

11.14. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	физкультура
2	физкультура & подтягивания & отжимания
3	физкультура & подтягивания
4	физкультура фитнес

Ответ: _____.

11.15. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке убывания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	алгоритм & (программа данные)
2	алгоритм & программа & данные
3	программа данные
4	алгоритм программа данные

Ответ: _____.

11.16. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке убывания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	алгоритм & программа данные
2	алгоритм & программа & данные
3	алгоритм данные
4	алгоритм программа данные

Ответ: _____.

11.17. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке убывания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	алгоритм программа & данные
2	алгоритм программа данные
3	программа & данные
4	алгоритм & программа & данные

Ответ: _____.

11.18. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке убывания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &.

1	рога & копыта хвосты
2	рога копыта хвосты
3	рога & копыта
4	рога & (копыта хвосты)

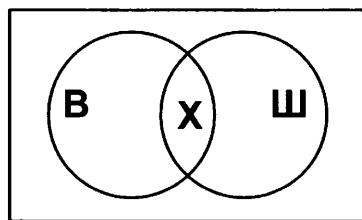
Ответ: _____.

Пример. В классе учатся 25 человек. Из них 17 человек ходят на кружок по волейболу и 13 человек ходят на шахматный кружок. Известно, что ни в один кружок не ходят 6 человек. Сколько человек занимаются и волейболом, и шахматами?

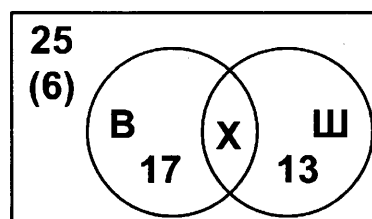
Решение:

Эта задача не относится к категории сетевых технологий. Но понимание ее решения требуется для решения новых заданий на количество запросов к поисковому серверу (которые будут рассмотрены ниже).

Нагляднее всего для решения воспользоваться диаграммой Эйлера:



Прямоугольником обозначено все множество учащихся класса. Кругами, помеченными буквами В и Ш, обозначены те учащиеся, которые занимаются волейболом и шахматами соответственно. На пересечении кругов находится то множество, размер которого мы должны определить (обозначено буквой Х). Напишем на диаграмме числовые значения, которые нам известны:



В скобках написано число учащихся из общего множества, которые не принадлежат множествам В и Ш. Выразим через X размеры множеств, изображенных на диаграмме. Тогда количество учащихся, которые ходят только на волейбол, будет равно $17 - X$. А количество учащихся, которые ходят только на шахматы, будет равно $13 - X$.

Из диаграммы видно, что число учащихся, которые занимаются хотя бы в одном кружке, с одной стороны, равно $25 - 6$. С другой стороны, равно $(17 - X) + X + (13 - X)$. Приравняем эти два выражения, раскроем скобки и выразим X. Получим $25 - 6 = (17 - X) + X + (13 - X) \Leftrightarrow \Leftrightarrow 19 = 17 - X + X + 13 - X \Leftrightarrow 19 = 30 - X \Leftrightarrow X = 11$.

Другой способ решения:

Вспользуемся формулой включений и исключений из комбинаторики:

$$B \mid \text{Ш} = B + \text{Ш} - B \& \text{Ш}.$$

В приведенной формуле имеется некоторая математическая неточность, связанная с тем, что в ней используются обозначения самих множеств, в то время как предполагается, что действия выполняются над мощностями множеств. Мы решили не добавлять в формулу обозначения мощностей множеств, чтобы не загромождать ее.

$B \mid \text{Ш}$ — объединение множеств В и Ш. В нашем случае объем этого множества равен $25 - 6 = 19$.

$B \& \text{Ш}$ — пересечение множеств В и Ш. В нашем случае это искомая величина. Обозначим ее за X.

Подставляем известные значения, получаем уравнение: $19 = 17 + 13 - X$. Отсюда находим $X = 11$.

Заметим, что для трех переменных (X, Y, Z) формула включений и исключений выглядит так: $X \mid Y \mid Z = X + Y + Z - (X \& Y + X \& Z + Y \& Z) + X \& Y \& Z$.

Пример. При попытке осуществить запросы к поисковому серверу было найдено следующее количество страниц по каждому запросу:

Запрос	Количество страниц
<i>грюмзики</i>	1300
<i>изьямые</i>	1900
<i>грюмзики & изьямые</i>	700

Сколько страниц будет найдено по запросу *грюмзики | изьямые*?

Решение:

По формуле включений и исключений:

$$\text{грюмзики} \mid \text{изьямые} = \text{грюмзики} + \text{изьямые} - \text{грюмзики} \& \text{изьямые}$$

После подстановки получаем: $\text{грюмзики} \mid \text{изьямые} = 1300 + 1900 - 700 = 2500$.

11.19.

а) При попытке осуществить запросы к поисковому серверу было найдено следующее количество страниц по каждому запросу:

Запрос	Количество страниц (в тысячах)
<i>математика</i>	4300
<i>физика математика</i>	6700
<i>физика & математика</i>	2100

Сколько страниц будет найдено по запросу *физика*?

Ответ: _____.

- б) При попытке осуществить запросы к поисковому серверу было найдено следующее количество страниц по каждому запросу:

Запрос	Количество страниц (в тысячах)
<i>клубника</i>	4600
<i>ежевика</i>	6300
<i>клубника ежевика</i>	8700

Сколько страниц будет найдено по запросу *клубника & ежевика*?

Ответ: _____.

- в) При попытке осуществить запросы к поисковому серверу было найдено следующее количество страниц по каждому запросу:

Запрос	Количество страниц (в тысячах)
<i>тыква</i>	3500
<i>арбуз</i>	2800
<i>тыква & арбуз</i>	1400
<i>тыква & дыня</i>	900
<i>арбуз & дыня</i>	1700
<i>тыква & арбуз & дыня</i>	500
<i>тыква арбуз дыня</i>	7200

Сколько страниц будет найдено по запросу *дыня*?

Ответ: _____.

- г) При попытке осуществить запросы к поисковому серверу было найдено следующее количество страниц по каждому запросу:

Запрос	Количество страниц (в тысячах)
<i>диван</i>	3400
<i>чемодан</i>	5200
<i>саквояж</i>	2700
<i>диван & чемодан</i>	1600
<i>диван & саквояж</i>	1200
<i>чемодан & саквояж</i>	1300
<i>диван чемодан саквояж</i>	8300

Сколько страниц будет найдено по запросу *диван & чемодан & саквояж*?

Ответ: _____.

Пример. По правилам адресации в сетях TCP/IP каждому компьютеру выдается уникальный номер — IP-адрес. При этом считается, что каждый компьютер находится в некоторой сети, размер которой задается при помощи так называемой *маски сети*. Записав в двоичном представлении IP-адрес компьютера и маску сети и осуществив между ними поразрядную конъюнкцию, можно получить адрес сети. Определите адрес сети, если известно, что:

IP-адрес компьютера: 10.0.63.18

Маска сети: 255.255.252.0

Ответ закодируйте при помощи таблицы:

A	B	C	D	E	F	G	H
10	0	15	60	100	56	1	156

При этом точки записывать не нужно.

Решение:

Для получения адреса сети нужно записать в двоичной системе друг под другом IP-адрес компьютера и маску сети. При этом нужно учитывать, что каждое из чисел IP-адреса и маски сети хранится в 8-разрядном двоичном виде. После этого нужно осуществить поразрядную конъюнкцию этих чисел. После чего записать результат снова в десятичном виде. Так как 255 — это 8 подряд идущих единиц в двоичной системе, то в результате поразрядной конъюнкции с любым 8-разрядным числом ответом будет это же число. Поэтому (так как в маске сети первые два числа равны 255) первые два числа искомого адреса будут такими же, как и в IP-адресе (10.0). Так как 0 в 8-разрядном представлении является числом с 8-ю нулями, то поразрядная конъюнкция с ним превращает любое число в ноль. Поэтому последняя часть адреса сети будет ноль (0). Остается вычислить только третью часть адреса. Переведем в двоичную систему числа 63 и 252 и запишем их в 8-разрядном виде друг под другом:

00111111

11111100

Осуществив поразрядную конъюнкцию, получим число 00111100_2 , которое равно 60. В результате получаем адрес сети: 10.0.60.0. Найдем в таблице значения 10, 0, 60 и 0 и выпишем соответствующие буквы в этом порядке.

Ответ: ABDB

11.20. По правилам адресации в сетях TCP/IP каждому компьютеру выдается уникальный номер — IP-адрес. При этом считается, что каждый компьютер находится в некоторой сети, размер которой задается при помощи так называемой *маски сети*. Записав в двоичном представлении IP-адрес компьютера и маску сети и осуществив между ними поразрядную конъюнкцию, можно получить адрес сети. Определите адрес сети, если известно, что:

IP-адрес компьютера: 10.100.65.219

Маска сети: 255.224.0.0

Ответ закодируйте при помощи таблицы:

A	B	C	D	E	F	G	H
10	0	65	119	100	96	192	156

При этом точки записывать не нужно.

Ответ: _____

Пример. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырех байтов, причем каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 117.191.88.37 адрес сети равен 117.191.80.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Решение:

В результате поразрядной конъюнкции число 88 превратилось в число 80. Переведем оба числа в двоичную систему счисления и запишем друг под другом.

$$88 = 1011000_2, 80 = 1010000_2.$$

1011000

1010000

Видим, что первые 3 цифры (101) остались неизменными, а четвертая цифра (1) превратилась в 0. Значит, именно в этом месте в маске стоит 0. Так как маска состоит из последовательности единиц, за которой стоят нули, то именно в этом месте единицы меняются на нули. Получается, что маска — это 1110000_2 . Однако этот ответ неверный, потому что нас спрашивают третий БАЙТ маски. А в нашей маске только 7 бит. Значит, нужно добавить еще один бит слева. Это должна быть 1, так как в маске сначала стоят единицы. Значит, окончательный ответ $11110000_2 = 240$.

11.21. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырех байтов, причем каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 117.191.59.37 адрес сети равен 117.191.48.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

11.22. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырех байтов, причем каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 193.212.179.92 адрес сети равен 193.212.160.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

11.23. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырех байтов, причем каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 57.137.106.197 адрес сети равен 57.137.64.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Ответы

1. Количество информации. Скорость передачи информации

1.1. в), е); 1.2. а), б), г), д); 1.3. 1; 1.4. 3; 1.5. 3; 1.6. 7; 1.7. 6; 1.8. 8; 1.9. 250; 1.10. 490; 1.11. 420; 1.12. 600; 1.13. 560; 1.14. 600; 1.15. 10; 1.16. 5; 1.17. 10; 1.18. 2048; 1.19. 128; 1.20. 25; 1.21. 40; 1.22. 32; 1.23. 8; 1.24. 37,5; 1.25. 3750; 1.26. 3; 1.27. 96; 1.28. 108; 1.29. 4; 1.30. 320; 1.31. 800; 1.32. 322; 1.33. 260; 1.34. A122; 1.35. A13; 1.36. B3; 1.37. 405; 1.38. 256; 1.39. 108; 1.40. 500; 1.41. 1000; 1.42. 600; 1.43. 500; 1.44. 900.

2. Системы счисления

2.1. Система счисления — б); алфавит системы счисления — г); мощность алфавита — а); основание системы счисления — д).

2.2. Для каждого числа позиционной системы счисления — вес цифры зависит от ее положения в числе.

2.3. В основание (в десятичной — в 10, в восьмеричной — в 8, ...)

2.4. N, 0, N-1; 2.5. 10_N ; 2.6. 0 и 1; 2.7. 6, A, F, A пятнадцать.

2.8. 52436 — разряды, 43210 — номера разрядов, 8 — основание системы счисления

2.9. $2C_{16}$ и $2D_{16}$

2.10.

Система счисления

10	2	8	16	3	4	5
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	10	2	2	2	2	2
3	11	3	3	10	3	3
4	100	4	4	11	10	4
5	101	5	5	12	11	10
6	110	6	6	20	12	11
7	111	7	7	21	13	12
8	1000	10	8	22	20	13
9	1001	11	9	100	21	14
10	1010	12	A	101	22	20
11	1011	13	B	102	23	21
12	1100	14	C	110	30	22
13	1101	15	D	111	31	23
14	1110	16	E	112	32	24
15	1111	17	F	120	33	30
16	10000	20	10	121	100	31
17	10001	21	11	122	101	32

2.11. 101000_2 , 220_3 , 40_8 , $C0_{16}$, 22000_4 , 100_7 . 2.12. 10011_2 , 2022_3 , 517_8 , $9FF_{16}$, 3113_4 , 46_7

2.13. 10000_2 , $C0_{16}$, 2222_3 , $10A_{16}$. 2.14. Пары меньше-большее:

$1111_2-11000_2$, 1112_3-2212_3 , 277_8-630_8 , $10D_{16}-E0F_{16}$, $10000_4-32100_4$, $1FF_{16}-F00_{16}$

2.15. 3, 1, 2 2.16. 3, 2, 1

2.17. $2314_5 = 2 \cdot 5^3 + 3 \cdot 5^2 + 1 \cdot 5^1 + 4 \cdot 5^0 = 250 + 75 + 5 + 4 = 334_{10}$

$1010011_2 = 2^6 + 2^4 + 2^1 + 1 = 64 + 16 + 2 + 1 = 83_{10}$

$2AB_{16} = 2 \cdot 16^2 + 10 \cdot 16^1 + 11 = 512 + 160 + 11 = 683_{10}$

2.18. $1010110_2 = 86_{10}$, $21020_3 = 195_{10}$, $526_8 = 342_{10}$, $3CE_{16} = 974_{10}$, $3021_4 = 201_{10}$, $256_7 = 139_{10}$

2.19. 6 целых и 5 в остатке. 2.20. 4 целых и 3 в остатке. 2.21. 0 целых и 7 в остатке

2.22. $53 = 110101_2$. 2.23. $202 = 1302_5$.

2.24. $57 = 111001_2$, $65 = 2102_3$, $335 = 517_8$, $202 = CA_{16}$, $198 = 3012_4$, $139 = 256_7$

- 2.25. 3, 1, 2. 2.26. 215₈. 2.27. 374₈, 126₈. 2.28. 4, 1 (группами по 4), 2, 3. 2.29. 6AD₁₆
 2.30. 3FC₁₆, 75E₁₆. 2.31. 3, 1, 2. 2.32. 111001100₂. 2.33. 110000011₂, 011010101₂.
 2.34. 3, 2 (из 4 разрядов, 4 разряда), 1. 2.35. 110100100111₂.
 2.36. 111001000011₂, 101000001100₂. 2.37. 4; 2.38. 4; 2.39. 1; 2.40. 3; 2.41. 3; 2.42. 4; 2.43. 6;
 2.44. 2; 2.45. 3; 2.46. 4; 2.47. 6; 2.48. 4; 2.49. 3; 2.50. 5, 21; 2.51. 6, 14, 22; 2.52. 7, 16, 25;
 2.53. 8, 33; 2.54. 5, 13, 21; 2.55. 3, 7, 21; 2.56. 4, 6, 8, 12, 24; 2.57. 11, 22; 2.58. 9, 27;
 2.59. 13, 26; 2.60. 4; 2.61. 3; 2.62. 5; 2.63. 4; 2.64. 4; 2.65. 121₈; 2.66. 1000101101₂;
 2.67. 1031₄; 2.68. 17B₁₆; 2.69. 2; 2.70. 4; 2.71. 2; 2.72. 3; 2.73. 1; 2.74. 1010111₂;
 2.75. 11100011₂; 2.76. 1111011₂; 2.77. 2; 2.78. 1; 2.79. 8; 2.80. 9; 2.81. 18;
 2.82. Для N=7: 1) 237; 2) 11101101₂, 22210₃, 1422₅; 3) 11111000₂; 4) 370₈, F8₁₆; 5) 427₈;
 6) 279; 7) D9₁₆; 8) 217; 9) 100122₃; 10) 1333₅;
 2.85. 2008; 2.86. 2013; 2.87. 2013; 2.88. 11; 2.89. 7; 2.90. 6; 2.91. 8; 2.92. 13; 2.93. 29;
 2.94. 99; 2.95. 28; 2.96. 58; 2.97. 60.

3. Кодирование информации

- 3.1. б); 3.2. все; 3.3. 60; 3.4. 512; 3.5. 384; 3.6. 4; 3.7. 1; 3.8. 4; 3.9. 375; 3.10. 3; 3.11. 3;
 3.12. 3; 3.13. 4; 3.14. 2; 3.15. 1; 3.16. а-5, б-6, в-1, г-4, д-2, е-3; 3.17. а-3, б-4, в-5, г-2,
 д-1; 3.18. 1; 3.19. 4; 3.20. 2; 3.21. 4; 3.22. 1; 3.23. 8; 3.24. 5; 3.25. 7; 3.26. 16; 3.27. 8; 3.28. 2;
 3.29. 256; 3.30. 65536; 3.31. 3; 3.32. 1536; 3.33. 256; 3.34. 320; 3.35. 8; 3.36. 48; 3.37. 2048;
 3.38. 1024; 3.39. 32; 3.40. 8; 3.41. 128; 3.42. 3; 3.43. 3; 3.44. 2; 3.45. УУУОУ; 3.46. НЛЛМ;
 3.47. ТПТП; 3.48. 1; 3.49. 3; 3.50. 112; 3.51. 350; 3.52. 10; 3.53. 15; 3.54. 50; 3.55. 80.

4. Основы логики

- 4.3. а) A = 1, B = 1; б) A = 0, B = 0; в) A = 1, B = 0; г) A = 1, B = 1.
 4.7. а) 1; б) 2; в) 4; г) 1. 4.8. а) 1; б) 3; в) 2; г) 4. 4.9. а) 4; б) 4; в) 2; г) 3. 4.10. а) 3; б) 3;
 в) 1; г) 2.
 4.11. а) 4; б) 3; в) 2; г) 4. 4.12. а) 3; б) 4; в) 1; г) 4. 4.13. а) 3; б) 15; в) 9; г) 8. 4.14. а) 2; б) 4;
 в) 2; г) 3.
 4.15. а) 0001; б) 1000; в) 1010; г) 1001. 4.16. а) 12; б) 1; в) 14; г) 2. 4.17. а) 10; б) 14; в) 16; г) 68.
 4.18. 4. 4.19. а) 2; б) 3; в) 4;
 4.20. 3; 4.21. 1; 4.22. 3; 4.23. 4; 4.24. 2; 4.25. 1; 4.27. 251; 4.28. 118; 4.29. 0; 4.30. 128.
 4.31. 12; 4.32. 14; 4.33. 18; 4.34. 13; 4.35. 11.

5. Алгоритмизация и программирование

- 5.1. а) 25; б) 8; в) 0; г) -4012. 5.2. а) +...+; б) ·+...++; в) +...+·; г) +++...+. 5.10. а) $n(n-1)/2$;
 б) $n(2-n)$; в) n ; г) $-(1+n) \cdot n/2$.
 5.14. а) 2; б) 1; в) 2; г) 3. 5.15. а) 2; б) 345; в) 411; г) 112. 5.16. а) ВААВ; б) ВАА. 5.17. а) 12211;
 б) 21211; в) 11211; г) 211121. 5.18. а) 2, А8, Н4; б) 2, G1, F3; в) 1, Н3; г) 1, В3.
 5.19. а) 513; б) 10; в) 4; г) -9. 5.20. а) 2; б) 1; в) 4; г) 3. 5.21. а) 1; б) 1; в) 1; г) 4. 5.24. а)

Область	Условие 1 ($y \geq 0$)	Условие 2 ($y \leq 1$)	Условие 3 ($y \leq 2 - x$)	Условие 4 ($y \geq x * x$)	Программа выведет	Область обраба- тывается верно
А	да	нет	—	—	—	нет
В	да	нет	—	—	—	нет
С	да	нет	—	—	—	нет
Д	да	нет	—	—	—	нет
Е	да	да	да	нет	не принадлежит	да
Г	да	да	да	да	принадлежит	да
Г	да	да	да	нет	не принадлежит	нет
Н	да	да	нет	—	—	нет
И	нет	—	—	—	—	нет
Ж	нет	—	—	—	—	нет

Программа:

```
if (y<=1) and (y>=0) and ((y>=x*x) or (y<=2-x) and (x>=0)) then
  write('принадлежит')
else
  write('не принадлежит')
```

5.24. б)

Область	Условие 1 ($x \geq -2$)	Условие 2 ($x \leq 7$)	Условие 3 ($x < 4$)	Программа выведет	Область обрабатывается верно
А	нет	—	—	—	нет
В	да	да	да	принадлежит	да
С	да	да	да	принадлежит	нет
Д	да	да	нет	не принадлежит	нет
Е	да	нет	—	—	нет

Программа:

```
if (x>=-2) and (x<=1) or (x>=4) and (x<=7) then
  write('принадлежит')
else
  write('не принадлежит')
```

5.26. 15; 5.27. 24; 5.28. 72;

5.29.

1) 9;

2) 9 (также верным является любое натуральное число, имеющее цифру 9);

3) Вместо строки "k:=9;" должна быть строка "k:=0;" (вместо 0 может быть любое отрицательное число), вместо строки "while N >= 10 do" должна быть строка "while N > 0 do" (также возможно исправление "while N >= 1 do");

5.30.

1) 5;

2) 53 (верным является любое натуральное число, имеющее наименьшую цифру не первую);

3) Вместо строки "while N > 9 do" должна быть строка "while N > 0 do" (также возможно исправление "while N >= 1 do");

5.31.

1) 22;

2) 1 (верным также является любое натуральное число, имеющее последнюю цифру 0, а первую больше 1);

3) Вместо строки "k:=N mod 10;" должна быть строка "k:=0;", вместо строки "while N > 1 do" должна быть строка "while N > 0 do" (также возможно исправление "while N >= 1 do");

5.32.

1) 0;

2) 10 (также верным является любое натуральное число, имеющее цифру 0);

3) Вместо строки "k:=0;" должна быть строка "k:=1;", вместо строки "k:=k*N mod 10;" должна быть строка "k:=k*(N mod 10);";

5.33. 60; 5.34. 47; 5.35. 28; 5.37. 8, 5, 4, 6, 8, 5, 4, 3, 2; 5.38. 4, 2, 3, 4; 5.39. 2, 4, 2, 6, 3;

5.40. 5, 3, 1, 1, 2, 0, 1; 5.41. 3; 5.42. 1; 5.43. 61; 5.44. 175; 5.45. 930; 5.46. 349;

5.47. 46; 5.48. 46; 5.49. 1100; 5.50. 58; 5.51. 1101; 5.52. 1110; 5.53. 13; 5.54. 21; 5.55. 20;

5.56. 28; 5.57. 60; 5.58. 65; 5.59. 44; 5.60. 77877; 5.61. а) 8; б) 788.

6. Информационные модели

6.1. 2; 6.2. 3; 6.3. 3; 6.4. 3,5; 6.5. 11; 6.6. 17:25; 6.7. 18:15; 6.8. 16:30; 6.9. 16:20; 6.10. 14:10; 6.11. 25; 6.12. 30; 6.13. 24.

7. Определение выигрышной стратегии игры (Анализ и построение дерева игры)

7.1. См. полное решение и ответ в тексте.

7.2. Выигрывает Ваня. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучках соответственно.

Стартовая позиция	I ход (Петя), все варианты хода	II ход (Вани), выигрышные ходы	III ход (Петя), все варианты хода	IV ход (Вани), выигрышные ходы
4,3	12,3	36,3	Выигрыш Вани	
	6,3	6,9	18,9	54,9
			8,9	8,27
			6,27	Выигрыш Вани
			6,11	6,33
	4,9	6,9	Те же варианты III–IV ходов	
	4,5	12,5	36,5	Выигрыш Вани
			14,5	42,5
			12,15	36,15
			12,7	36,7

Таблица содержит все возможные варианты ходов первого игрока (Петя). Из нее видно, что при любом ходе первого игрока у второго (Вани) имеется ход, приводящий к победе.

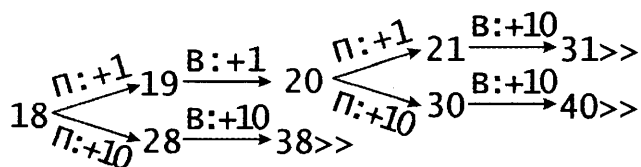
7.4. 1а) $S = 21 \dots 30$. Петя должен добавить в кучу 10 камней и выиграть. Для $S < 20$ невозможно одним ходом получить в куче больше 30 камней.

1б) $S = 20$. Куда бы ни ходил Петя ($20 + 1 = 21$ или $20 + 10 = 30$), Ваня должен добавить в кучу 10 камней и выиграть.

2. $S = 10$ и $S = 19$. Своим первым ходом Петя должен получить в куче 20 камней ($10 + 10 = 20$ или $19 + 1 = 20$). При любом ответном ходе Вани ($20 + 1 = 21$ или $20 + 10 = 30$) Ваня должен добавить в кучу 10 камней и выиграть.

3. $S = 18$. Если Петя добавит в кучу 10 камней ($18 + 10 = 28$), Ваня тоже добавит в кучу 10 камней и выиграет ($28 + 10 = 38$).

Если же Петя добавит в кучу 1 камень ($18 + 1 = 19$), Ваня тоже добавит в кучу 1 камень ($19 + 1 = 20$). При любом ответном ходе Пети ($20 + 1 = 21$ или $20 + 10 = 30$) Ваня должен добавить в кучу 10 камней и выиграть. Рассмотрим дерево игры:



В этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня – только ход, соответствующий стратегии, которую выбрал Ваня.

7.6.

1а. $S = 29 \dots 57$. Петя должен удвоить число камней во второй куче.

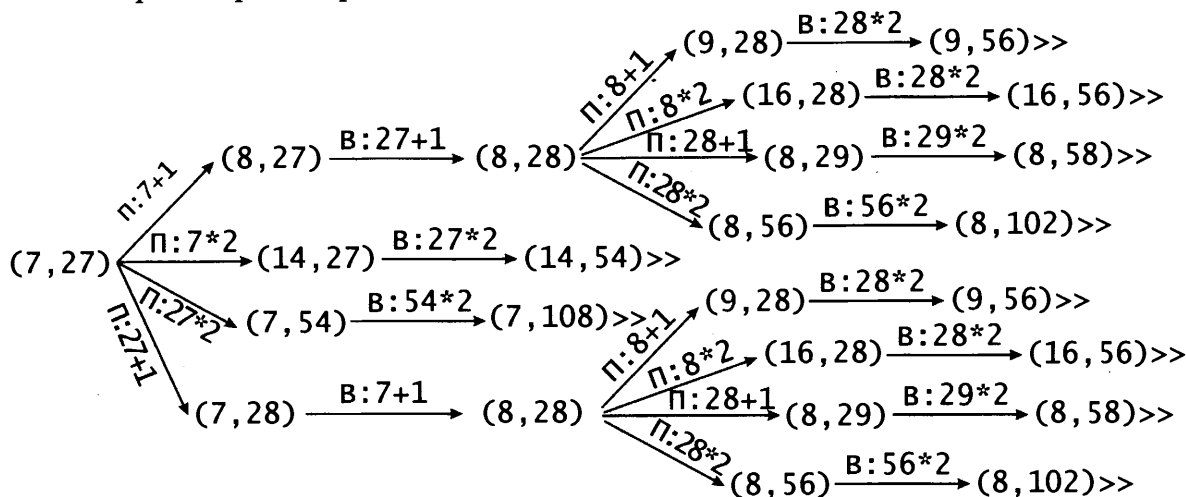
1б. Единственный подходящий вариант ($S = 28$) не подходит, т.к. при первом ходе Пети «+1 к первой куче» получается ситуация (8, 28), из которой нельзя выиграть одним ходом. Числа, меньшие 28, тоже не подходят (при $S = 27$ добавление 1 камня во вторую

кучу приводит к (7, 28), из которой не выиграть). Аналогично, можно такое показать для каждого меньшего S.

2. S = 28. Первым ходом Петя должен добавить 1 камень в первую кучу (получится (8, 28)). При любом ответном ходе Вани Петя должен удвоить вторую кучу и выиграть.

3. S = 27.

Рассмотрим дерево игры:



В дереве рассмотрены все возможные ходы Пети и ответные ходы Вани, соответствующие выигрышной стратегии. Знаком ">>" обозначены позиции, в которых партия заканчивается.

7.8.

Задание 1.

Из позиции (10, 26) можно получить (11, 26), (20, 26), (10, 27), (10, 52). Во всех случаях сумма камней в кучах меньше 63, однако удвоение второй кучи приводит к выигрышу (в сумме будет не менее 63 камней). Значит, позиция (10, 26) — проигрышная и выигрышную стратегию имеет Витя. Он выигрывает своим первым ходом.

Из позиции (12, 25) можно получить (13, 25), (24, 25), (12, 26), (12, 50). Во всех этих случаях сумма камней в кучах также меньше 63 и удвоение второй кучи приводит к выигрышу. Значит, позиция (12, 25) — также проигрышная и выигрышную стратегию имеет Витя. Он выигрывает своим первым ходом.

Задание 2.

Для каждой из рассматриваемых позиций существует ход, который приводит в проигрышные позиции, рассмотренные в задании 1.

(6, 25) → (12, 25); (9, 26) → (10, 26); (10, 25) → (10, 26).

Значит, эти позиции — выигрышные. Выигрышную стратегию имеет Паша. Его первый ход нами указан. Паша выигрывает на своем втором ходу. Игра из позиций (10, 26) и (12, 25) рассмотрена в задании 1.

Задание 3.

В позиции (9, 25) выигрышную стратегию имеет Витя. Если Паша первым ходом пойдет в (10, 25) или (9, 26), Витя пойдет в (10, 26) и выиграет своим вторым ходом. Стратегия его игры описана в заданиях 1 и 2.

Если же Паша первым ходом пойдет в (18, 25) или (9, 50), Витя выиграет своим первым ходом, удвоив количество камней во второй куче.

8. Файловая система компьютера

8.5. а) 3; б) 1. 8.6. а) 3; б) 4. 8.7. а) 3; б) 4; в) 2; г) 3. 8.8. а) 3; б) 3; в) 2; г) 4. 8.9. а) 2; б) 4; в) 3; г) 4.

9. Электронные таблицы

9.1. все; 9.2. а), б), г), д), з); 9.3. б), в), д), з), к); 9.4. а-3, б-9, в-5, г-11, д-1, е-10, ж-4, з-7, и-6, к-8, л-2; 9.5. а), б), г), д), е), ж), з), и) (т.е., все, кроме в); 9.6. а-4, б-5, в-1, г-3, д-2; 9.7. а) 1, б) 3, в) 4, г) 12, д) 9, е) 4, ж) 18, з) 256; 9.8. а) 0, б) 7, в) 10, г) 2,5, д) 6; 9.9. 4; 9.10. 2; 9.11. 1; 9.12. -11; 9.13. 8; 9.14. 52; 9.15. $=\$B2+D\2 ; 9.16. $=\$C1-A\3 ; 9.17. $=\$A\$4+D1$; 9.18. $=2 * \$A2$; 9.19. 2; 9.20. 4; 9.21. 1; 9.22. 1; 9.23. а), в); 9.24. 1; 9.25. б), в); 9.26. 6; 9.27. 8. 9.28. 3; 9.29. 4; 9.30. 3.

10. Базы данных

10.1. все; 10.2. а-4, б-6, в-3, г-5, д-1, е-2; 10.3. 3; 10.5. 5; 10.7. 4; 10.8 а-2, б-3, в-1; 10.9. бгва; 10.10. 1; 10.11. 3; 10.12. 2; 10.13. 4; 10.14. 1; 10.15. 3; 10.16. 3; 10.17. 4; 10.18. 210; 10.19. 205; 10.20. 1; 10.21. 3; 10.22. 2; 10.23. 4.

11. Сетевые технологии

11.1. а), б) в), г), е), ж), з); 11.2. 3; 11.3. ГБВА; 11.4. БВГА; 11.5. БАГВ; 11.6. все; 11.7. ЖГВБАЕД; 11.8. ГЖАВДЕБ; 11.9. ДВГАЕБЖ; 11.10. ЖГДВБЕА; 11.11. 321; 11.12. 2341; 11.13. 1234; 11.14. 2314; 11.15. 4312; 11.16. 4312; 11.17. 2134; 11.18. 2143; 11.19. а) 4500; б) 2200; в) 4400; г) 1100. 11.20. АFBV; 11.21. 240; 11.22. 224; 11.23. 192.

Вариант 1

Часть 1

Ответом к заданиям 1–23 является число, последовательность букв или цифр, которые следует записать в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов и каких-либо дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведенными образцами.

1. Сколько существует натуральных чисел x , для которых выполнено неравенство $332_8 < x < EA_{16}$?

В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____ .

2. Логическая функция F задаётся выражением $a \vee \neg b \vee (\neg c \wedge d)$.

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F ложна.

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных a, b, c, d .

Перем. 1	Перем. 2	Перем. 3	Перем. 4	Функция
???	???	???	???	F
0	0	1	0	0
1	0	1	0	0
1	0	1	1	0

В ответе напишите буквы a, b, c, d в том порядке, в котором идут соответствующие им столбцы (сначала — буква, соответствующая первому столбцу; затем — буква, соответствующая второму столбцу, и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Если бы функция была задана выражением $\neg a \vee b$, зависящим от двух переменных: a и b и был приведён фрагмент её таблицы истинности, содержащий все наборы аргументов, при которых функция F истинна.

Перем. 1	Перем. 2	Функция
???	???	F
0	0	1
1	0	1
1	1	1

Тогда первому столбцу соответствовала бы переменная b , а второму столбцу — переменная a . В ответе следовало бы написать: ba .

Ответ: _____ .

3. Между населенными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице (если ячейка пуста — дороги нет).

	А	В	С	D	E	F
А		7	3			
В	7		2	4	1	
С	3	2		7	5	9
D		4	7		2	3
E		1	5	2		7
F			9	3	7	

Определите длину кратчайшего пути между пунктами А и F.

Ответ: _____ .

4. Во фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных числовой идентификатор (ID) внука Дени К.Д.

Таблица 1

ID	Фамилия И.О.	Пол
42	Бабенко А.Е.	М
82	Сорт А.К.	Ж
53	Марченко Е.Е.	М
21	Ирга О.Б.	Ж
26	Гано И.Е.	Ж
71	Камо Е.А.	М
14	Дени К.Д.	Ж
34	Ивенко Д.Я.	М
78	Исава Г.А.	Ж
57	Арне А.А.	М

Таблица 2

ID Родителя	ID Ребенка
53	42
21	42
71	53
14	53
34	14
78	14
71	26
14	26
26	82
57	78

Ответ: _____ .

5. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приемной стороне канала связи. Использовали код: А-00, Б-01, В-100, Г-101, Д-110. Укажите кратчайшее кодовое слово для буквы Д, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.

Ответ: _____ .

6. На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число V следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа еще два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа V .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 57. В ответе это число запишите в двоичной системе счисления.

Ответ: _____ .

7. Дан фрагмент электронной таблицы

	A	B	C	D
1	8	3	7	20
2	13	=A2+\$C\$1	25	41
3	2	17		50

Формулу из ячейки B2 скопировали в ячейку C3 так, что числовое значение ячейки C3 стало отличаться от числового значения ячейки B2. Каково стало числовое значение ячейки C3?

Ответ: _____ .

8. Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre>DIM S, N AS INTEGER S = 0 N = 0 WHILE S <= 95 S = S + 10 N = N + 5 WEND PRINT N</pre>	<pre>s = 0 n = 0 while s <= 95: s = s + 10 n = n + 5 print(n)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач цел n, s n := 0 s := 0 нц пока s <= 95 s := s + 10 n := n + 5 кц вывод n кон</pre>	<pre>var s, n: integer; begin s := 0; n := 0; while s <= 95 do begin s := s + 10; n := n + 5; end; writeln(n) end.</pre>
Си	
<pre>#include<stdio.h> int main() { int s = 0, n = 0; while (s <= 95) { s = s + 10; n = n + 5; } printf("%d\n", n); return 0; }</pre>	

Ответ: _____ .

9. Какой минимальный объем памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 1024×512 пикселей при условии, что в изображении могут использоваться 8 различных цветов? В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____ .

10. Андрей составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует свое кодовое слово. В качестве кодовых слов Андрей использует 5-буквенные слова, в которых есть только буквы О, Р, Е, Л, причем буква Р появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Андрей?

Ответ: _____ .

11. Ниже на пяти языках программирования записана рекурсивная функция (процедура) F

Паскаль	Си
<pre> procedure f(n:integer); begin writeln(n); if n < 5 then begin f(n+1); f(n+2) end end; </pre>	<pre> #include <stdio.h> void f(int n) { printf("%d",n); if(n<5) { f(n+1); f(n+2); } } </pre>
Бейсик	Алгоритмический язык
<pre> FUNCTION F(N) PRINT N IF N < 5 THEN F(N+1) F(N+2) END IF END FUNCTION </pre>	<pre> алг цел А(цел n) нач вывод n если n < 5 то F(n+1) F(n+2) все кон </pre>
Python	
<pre> def F(n): print(n) if n < 5: F(n + 1) F(n + 2) </pre>	

Чему равна сумма выводимых на экран чисел при вызове F(1)?

Ответ: _____ .

12. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 251.137.219.94

Маска: 255.255.248.0

Выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в качестве ответа в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	94	137	216	219	248	251	255

Пример.

Пусть искомый IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

Ответ: _____ .

13. Репетиционный экзамен сдают 9 потоков по 100 человек в каждом. Каждому из них выделяют специальный код, состоящий из номера потока и номера в потоке. При кодировании этих номеров участников проверяющая система использует минимально возможное количество бит, одинаковое для каждого участника, отдельно для номера потока и номера в потоке. При этом для записи кода используется минимально возможное и одинаково целое количество байтов. Каков объем информации в байтах, записанный устройством после регистрации 80 участников? В ответе укажите только число.

Ответ: _____ .

14. Исполнитель Редактор получает на вход строку цифр и преобразовывает ее. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) **заменить** (v , w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды

заменить (222, 34)

преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v , то выполнение команды **заменить** (v , w) не меняет эту строку.

Б) **нашлось** (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

ТО команда1

ИНАЧЕ команда2

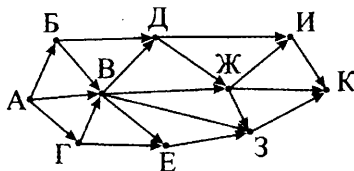
КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно).
 Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 80 идущих подряд цифр 7? В ответе запишите полученную строку.

НАЧАЛО
 ПОКА нашлось (777) ИЛИ нашлось (888)
 ЕСЛИ нашлось (777)
 ТО заменить (777, 8)
 ИНАЧЕ заменить (888, 7)
 КОНЕЦ ЕСЛИ
 КОНЕЦ ПОКА
 КОНЕЦ

Ответ: _____.

15. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Ответ: _____.

16. Значение арифметического выражения: $25^{10} + 5^8 - 125$ — записали в системе счисления с основанием 5. Сколько цифр «4» содержится в этой записи?

Ответ: _____.

17. В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

Запрос	Количество найденных страниц (в тысячах)
Яблоки & Сливы & Груши	3
Яблоки & Груши	10
Яблоки Сливы	60
Сливы Груши	50
Яблоки Сливы Груши	80

Какое количество страниц (в тысячах) будет найдено по запросу: Сливы?

Ответ: _____.

18. На числовой прямой даны два отрезка: $P = [12, 18]$ и $Q = [10, 24]$.
 Укажите наибольшую возможную длину отрезка А, для которого формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x .

Ответ: _____.

19. В программе описан одномерный целочисленный массив с индексами от 1 до 100. Ниже представлен фрагмент программы, обрабатывающей данный массив.

Паскаль	Си
<pre>k:=1; min:=A[1]; for i:=2 to 100 do if A[i]<min then k:=i;</pre>	<pre>k = 1; min = A[1]; for (i=2 ; i<=100 ; i++) if(A[i]<min) k=i;</pre>
Бейсик	Алгоритмический язык
<pre>k = 1 min = A(1) FOR i=2 TO 100 IF A(i) < min THEN k = i ENDIF NEXT i</pre>	<pre>k:=1 min:=A[1] НЦ ДЛЯ i ОТ 2 ДО 100 ЕСЛИ A[i]<min ТО k:=i ВСЕ КЦ</pre>
Python	
<pre>k = 1 min = A[1] for i in range(2,101): if A[i] < min: k=i</pre>	

Чему будет равно значение переменной k после выполнения данной программы, если известно, что массив был заполнен четными двузначными числами, расположенными в порядке неубывания от начала к концу массива?

Ответ: _____ .

20. Получив на вход число x , алгоритм печатает два числа L и M . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 4, а потом 8.

Паскаль	Си
<pre>var x, L, M: integer; begin readln(x); L := 0; M := 0; while x>0 do begin M := M+1; if x mod 2 <> 0 then begin L := L + 1; end; x := x div 2; end; writeln(L); write(M); end.</pre>	<pre>#include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L = 0; M = 0; while (x > 0) { M = M + 1; if(x % 2 != 0) { L = L + 1; } x = x / 2; } printf("%d\n%d", L, M); }</pre>

Бейсик	Алгоритмический язык
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X > 0 M = M + 1 IF X MOD 2 <> 0 THEN L = L + 1 ENDIF X = X \ 2 WEND PRINT L PRINT M </pre>	<pre> алг нач цел x, L, M ВВОД x L := 0 M := 0 нц пока x > 0 M := M + 1 если mod(x,2)<>0 то L := L + 1 все x := div(x,2) кц ВЫВОД L, <u>нс</u>, M кон </pre>
Python	
<pre> x = int(input()) L = 0 M = 0 while x>0 do : M = M+1 if x % 2 <> 0 : L=L+1 x = x // 2 print(L) ptint(M) </pre>	

Ответ: _____ .

21. Ниже на пяти языках приведена программа. Определить, при скольких различных значениях k программа печатает тот же результат, что и при $k = 90$ (включая $k = 90$).

Паскаль	Си
<pre> function f(n:integer):integer; begin if n = 0 then f := 0 else f := n+n-1 + f(n-1) end; var k,i :integer; BEGIN readln(k); i:=12; while f(i)>k do i:=i-1; write(i); END. </pre>	<pre> int f(int n) { if (n == 0) return 0; else return n+n-1 + f(n-1); } void main() { int k, i; scanf("%d",&k); i = 12; while (f(i)>k) i--; printf("%d", i); } </pre>

Бейсик	Алгоритмический язык
<pre> DIM K, I AS INTEGER INPUT K I = 12 WHILE F(I) > K I = I - 1 WEND PRINT I FUNCTION F(N) IF N = 0 F = 0 ELSE F = N+N-1 + F(N-1) END FUNCTION </pre>	<pre> алг нач цел k, i ввод k i := 12 нц пока F(i) > k i := i - 1 кц вывод i кон алг цел F(цел n) нач если n = 0 то знач := 0 иначе знач := n+n-1 + F(n-1) все кон </pre>
Python	
<pre> def f(n): if n == 0 : return 0 else : return n+n-1 + f(n-1) k = int(input()) i = 12 while f(i)>k : i-- print(i) </pre>	

Ответ: _____ .

22. Исполнитель Калькулятор преобразует число, записанное на экране.

У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1,
2. Прибавить 2,
3. Умножить на 2.

Первая из них увеличивает число на экране на 1, вторая увеличивает его на 2, третья умножает его на 2.

Программа для исполнителя Калькулятор — это последовательность команд.

Сколько существует таких программ, которые исходное число 2 преобразуют в число 10, и при этом траектория вычислений программы содержит число 5?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 123 при исходном числе 1 траектория будет состоять из чисел 2, 4, 8.

Ответ: _____ .

23. Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) = 1$$

$$(\neg y_1 \vee y_2) \wedge (\neg y_2 \vee y_3) \wedge (\neg y_3 \vee y_4) \wedge (\neg y_4 \vee y_5) = 1$$

$$(y_1 \rightarrow x_1) \wedge (y_2 \rightarrow x_2) \wedge (y_3 \rightarrow x_3) \wedge (y_4 \rightarrow x_4) \wedge (y_5 \rightarrow x_5) = 1$$

Ответ: _____ .

Часть 2

Для записи ответов на задания этой части (24–27) используйте бланк ответов № 2. Запишите сначала номер задания (24, 25 и т.д.), а затем полное решение. Ответы записывайте четко и разборчиво.

24. Требовалось написать программу, которая считывает с клавиатуры n натуральных чисел (например, $n = 4$), каждое из которых не больше 1000, и выводит на экран минимальное нечетное из них. Если таких чисел нет, программа должна вывести на экран «NO».

Программист торопился и написал программу неправильно.

Паскаль	Си
<pre> const n = 4; var i, x : integer; min, k : integer; begin k := 0; min := 0; for i := 1 to n do begin read(x); if x mod 2 <> 0 then begin k := k + 1; if min < x then min := x end end; end; if k > 0 then writeln(min) else writeln('NO') end. </pre>	<pre> #include <stdio.h> #define n 4 void main(void) { int i, x; int min, k; k = 0; min = 0; for (i = 1; i <= n; i++) { scanf("%d", &x); if (x % 2 != 0) { k++; if (min < x) min = x; } } if (k > 0) printf("%d\n", min); else printf("NO\n"); } </pre>

Бейсик	Алгоритмический язык
<pre> CONST n = 4 k = 0 min = 0 FOR I = 1 TO n INPUT x IF x mod 2 <> 0 THEN k = k + 1 IF min < x THEN min = x END IF END IF NEXT I IF k > 0 THEN PRINT min ELSE PRINT "NO" END IF </pre>	<pre> алг нач цел n = 4 цел i, x цел min, k k := 0 min := 0 нц для i от 1 до n ввод x если mod(x, 2) <> 0 то k := k + 1 если min < x то min := x все все кц если k > 0 то вывод min иначе вывод "NO" все кон </pre>
Python	
<pre> n = 4 k = 0; min = 0; for i in range(1, n + 1): x = int(input()) if x % 2 != 0 : k++ if min < x : min = x if k > 0 : print(min) else : print("NO") } </pre>	

Последовательно выполните следующее.

1. Укажите, что выведет на экран программа при вводе последовательности чисел 1 2 3 4.
 2. Приведите пример такой последовательности из n допустимых чисел, при котором программа работает верно.
 3. Исправьте все ошибки в программе. Для этого для каждой ошибки приведите строку, которая написана неверно, и строку, на которую ее нужно заменить, чтобы программа работала верно.
25. Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 1 до 99. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество элементов массива, сумма цифр которых не делится на 3.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Си
<pre>const N=40; var a: array [1..N] of integer; i, j, k, s: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>#include <stdio.h> #define N 40 void main(void) {int a[N]; int i, j, k, s; for (i=0; i<N; i++) scanf("%f", &a[i]); ... }</pre>
Бейсик	Алгоритмический язык
<pre>N=40 DIM A(N) AS INTEGER DIM I, J, K, S AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N=40 <u>целтаб</u> a[1:N] <u>цел</u> i, j, k, s <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Python	
<pre>N = 40 i = None j = None k = None a = [int(input()) for i in range(N)] ...</pre>	

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия.

26. Два игрока, Паша и Витя, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырех позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 61. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 61 или больше камней.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он

должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (3, 29) и (1, 30) выигрышная стратегия есть у Паши. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций (4, 28), (6, 27) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (4, 27), (5, 27), (6, 26) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (5, 26) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведет к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

27. На вход программе подается набор символов, заканчивающийся символом 0 (в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введен ноль, или считывать данные из файла). Ноль в этом наборе единственный. Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например Borland Pascal 7,0), которая будет составлять из всех имеющихся цифр (кроме завершающей последовательности нуля) минимальное число, состоящее ровно из трех повторяющихся одинаковых групп цифр ненулевой длины. Составленное число следует вывести на экран или в файл.

В случае невозможности составить такое число программа должна вывести «NO».

Например, пусть на вход подаются следующие символы:

fd7s22hg 547h2j 47x5 540

В данном случае программа должна вывести:

245724572457

Вариант 2

Часть 1

Ответом к заданиям 1–23 является число, последовательность букв или цифр, которые следует записать в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов и каких-либо дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведенными образцами.

1. Сколько существует натуральных чисел x , для которых выполнено неравенство $101011_2 < x < 75_8$?

В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____ .

2. Логическая функция F задаётся выражением $\neg a \vee b \vee (c \wedge \neg d)$.

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F ложна.

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных a, b, c, d .

Перем. 1	Перем. 2	Перем. 3	Перем. 4	Функция
???	???	???	???	F
0	0	1	0	0
1	0	1	0	0
1	0	1	1	0

В ответе напишите буквы a, b, c, d в том порядке, в котором идут соответствующие им столбцы (сначала — буква, соответствующая первому столбцу; затем — буква, соответствующая второму столбцу, и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Если бы функция была задана выражением $\neg a \vee b$, зависящим от двух переменных: a и b и был приведён фрагмент её таблицы истинности, содержащий все наборы аргументов, при которых функция F истинна,

Перем. 1	Перем. 2	Функция
???	???	F
0	0	1
1	0	1
1	1	1

тогда первому столбцу соответствовала бы переменная b , а второму столбцу — переменная a . В ответе следовало бы написать: ba .

Ответ: _____ .

3. Между населенными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице (если ячейка пуста — дороги нет).

	А	В	С	D	E	F
А		7	2			
В	7		4	2	2	4
С	2	4		5	1	
D		2	5		4	3
E		2	1	4		8
F		4		3	8	

Определите длину кратчайшего пути между пунктами А и F.

Ответ: _____ .

4. Во фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных числовой идентификатор (ID) бабушки Иваненко И.Л.

Таблица 1

ID	Фамилия И.О.	Пол
34	Арно В.А.	Ж
18	Жуйко А.И.	М
61	Кирта У.Т.	Ж
96	Вирк Я.М.	Ж
47	Иваненко И.Л.	М
29	Калпен Л.В.	М
56	Ирита Е.О.	Ж
84	Кименко З.Т.	Ж
72	Олинг К.Л.	Ж
15	Гираб В.А.	М

Таблица 2

ID_Родителя	ID_Ребенка
18	34
61	34
96	18
47	18
29	47
56	47
84	29
15	29
29	72
56	72

Ответ: _____ .

5. По каналу связи передаются сообщения, содержащие только 4 буквы А, Б, В, Г; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв А, Б, В используются такие кодовые слова: А: 000, Б: 1, В: 001. Укажите кратчайшее кодовое слово для буквы Г, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.

Ответ: _____ .

6. Автомат получает на вход четырехзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвертая цифры исходного числа.
2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 4853. Суммы: $4 + 8 = 12$; $5 + 3 = 8$. Результат: 812.

Укажите наибольшее число, в результате обработки которого автомат выдаст число 1517.

Ответ: _____.

7. Дан фрагмент электронной таблицы

	A	B	C	D
1	8	3	9	20
2	13	=\$A2+\$C1	25	41
3	2	17		50

Формулу из ячейки B2 скопировали в ячейку C3 так, что числовое значение ячейки C3 стало отличаться от числового значения ячейки B2. Каково стало числовое значение ячейки C3?

Ответ: _____.

8. Определите, что будет напечатано в результате выполнения программы:

Паскаль	Си
<pre>var i,s:integer; begin i:=1; s:=105; while s > 5 do begin s := s - 2; i := i + 1 end; write(i); end.</pre>	<pre>#include <stdio.h> void main(void) { int i,s; i = 1; s = 105; while (s > 5) { s = s - 2; i = i + 1; } printf("%d",i); }</pre>
Бейсик	Алгоритмический язык
<pre>DIM I,S AS INTEGER I = 1 S = 105 WHILE S > 5 S = S - 2 I = I + 1 WEND PRINT I</pre>	<pre>алг нач цел i, s i := 1 s := 105 нц пока s > 5 s := s - 2 i := i + 1 кц вывод i кон</pre>
Python	
<pre>i = 1 s = 105 while s > 5 : s = s - 2 i = i + 1 print(i)</pre>	

Ответ: _____.

9. Производится одноканальная (моно) звукозапись с частотой дискретизации 4 кГц и 32-битным разрешением. Запись длится 10 минут, ее результаты записываются в файл, сжатие данных не производится. Определите приблизительно размер получившегося файла (в Мбайтах). В качестве ответа укажите ближайшее к размеру файла целое число, кратное 3.

Ответ: _____.

10. Все 4-буквенные слова, составленные из букв П, Р, С, Т, записаны в алфавитном порядке.

Вот начало списка:

1. ПППП
2. ПППР
3. ПППС
4. ПППТ
5. ППРП

.....

На каком месте в списке стоит первое слово, начинающееся с буквы Р?

Ответ: _____.

11. Ниже на пяти языках программирования записана рекурсивная функция (процедура) F.

Паскаль	Си
<pre> procedure f(n:integer); begin writeln(n); if n > 3 then begin f(n-1); f(n-3) end end; end; </pre>	<pre> #include <stdio.h> void f(int n) { printf("%d",n); if(n > 3) { f(n-1); f(n-3); } } </pre>
Бейсик	Алгоритмический язык
<pre> FUNCTION F(N) PRINT N IF N > 3 THEN F(N-1) F(N-3) END IF END FUNCTION </pre>	<pre> алг F(цел n) нач вывод n если n > 3 то F(n-1) F(n-3) все кон </pre>
Python	
<pre> def F(n): print(n) if n > 3: F(n - 1) F(n - 3) </pre>	

Определите сумму чисел, которые выведет программа при вызове F(5).

Ответ: _____.

12. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP — адрес узла: 153.236.189.51

Маска: 255.255.224.0

Выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в качестве ответа в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	51	153	160	189	224	236	255

Пример.

Пусть искомый IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

Ответ: _____.

13. Репетиционный экзамен в СПб сдают 20 потоков по 50 человек в каждом. Каждому из них выделяют специальный код, состоящий из номера потока и номера в потоке. При кодировании этих номеров участников проверяющая система использует минимально возможное количество бит, одинаковое для каждого участника, отдельно для номера потока и номера в потоке. При этом для записи кода используется минимально возможное и одинаково целое количество байтов. Каков объем информации в байтах, записанный устройством после регистрации 70 участников? В ответе укажите только число.

Ответ: _____.

14. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, включает 4 команды-приказа и 4 команды проверки условия.

Команды-приказы:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Если РОБОТ начнет движение в сторону находящейся рядом с ним стены, то он разрушится, и программа прервется.

Другие четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

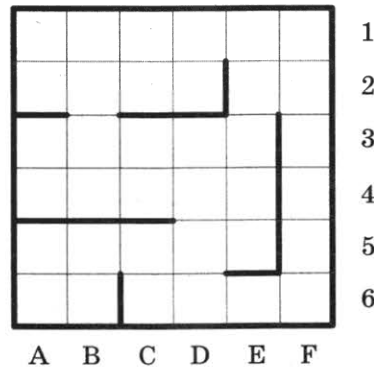
сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл:
ПОКА < условие >
 последовательность команд
КОНЕЦ ПОКА
выполняется, пока условие истинно.
В конструкции
ЕСЛИ < условие >
 ТО команда1
 ИНАЧЕ команда2
КОНЕЦ ЕСЛИ

выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

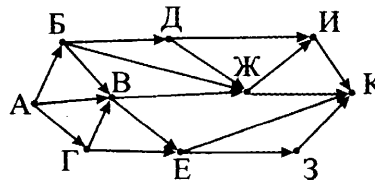
Сколько клеток приведенного лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную ниже программу, РОБОТ уцелеет и остановится в клетке F6?

НАЧАЛО
ПОКА < снизу свободно ИЛИ справа свободно >
 ЕСЛИ < справа свободно > **ТО** **вправо** **КОНЕЦ ЕСЛИ**
 ПОКА < снизу свободно > **вниз** **КОНЕЦ ПОКА**
КОНЕЦ ПОКА
КОНЕЦ



Ответ: _____.

15. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Ответ: _____.

16. Значение арифметического выражения: $9^{2016} + 3^{2015} - 2$ — записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Ответ: _____.

17. В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

Запрос	Количество найденных страниц (в тысячах)
Арбузы	84
Дыни	47
Тыквы	19
Арбузы & Тыквы	12
Тыквы & Дыни	0
Арбузы Тыквы Дыни	120

Какое количество страниц (в тысячах) будет найдено по запросу: *Арбузы & Дыни*?

Ответ: _____ .

18. Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $11 \& 6 = 1011_2 \& 0110_2 = 0010_2 = 2$.

Для какого наибольшего неотрицательного целого числа A формула

$$x \& A \neq 0 \rightarrow (x \& 9 = 0 \rightarrow x \& 3 \neq 0)$$

тождественно истинна (т.е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____ .

19. В программе описан одномерный целочисленный массив с индексами от 1 до 100. Ниже представлен фрагмент программы, обрабатывающей данный массив.

Паскаль	Си
<pre>k:=1; for i:=2 to 100 do if A[i]>=A[k] then k:= i;</pre>	<pre>k = 1 for (i=2 ; i<=100 ; i++) if(A[i]>=A[k]) k=i;</pre>
Бейсик	Алгоритмический язык
<pre>k = 1 FOR i=2 TO 100 IF A(i) >= A[k] THEN k = i ENDIF NEXT i</pre>	<pre>k:=1 нц для i от 2 до 100 если A[i]>=A[k] то k:= i все кц</pre>
Python	
<pre>k = 1 for i in range(2,101): if A[i]>=A[k]: k=i</pre>	

Чему будет равно значение переменной k после выполнения данной программы, если известно, что массив был заполнен четными двузначными числами, расположенными в порядке неубывания от начала к концу массива?

Ответ: _____ .

20. Получив на вход число x , алгоритм печатает два числа L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 5, а потом 8.

Паскаль	Си
<pre>var x, L, M: integer; begin readln(x); L := 0; M := 0; while x>0 do begin M := M+1; if x mod 2 = 0 then begin L := L + 1; end; x := x div 2; end; writeln(L); write(M); end.</pre>	<pre>#include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L = 0; M = 0; while (x > 0) { M = M + 1; if(x % 2 == 0) { L = L + 1; } x = x / 2; } printf("%d\n%d", L, M); }</pre>
Бейсик	Алгоритмический язык
<pre>DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X > 0 M = M + 1 IF X MOD 2 = 0 THEN L = L + 1 ENDIF X = X \ 2 WEND PRINT L PRINT M</pre>	<pre>алг нач цел x, L, M ввод x L := 0 M := 0 нц пока x > 0 M := M + 1 если mod(x,2)=0 то L := L + 1 все x := div(x,2) кц вывод L, M кон</pre>
Python	
<pre>x = int(input()) L = 0 M = 0 while x > 0 : M = M + 1 if x%2 == 0 : L = L + 1 x = x // 2 print(L,M)</pre>	

Ответ: _____.

21. Ниже на пяти языках приведена программа. Определить, при скольких различных значениях k программа печатает тот же результат, что и при $k = 60$ (включая $k = 60$).

Паскаль	Си
<pre>function f(n:integer):integer; var i,s:integer; begin s:=0; for i:=1 to n do s:=s+n; f:=s end; var k,i :integer; BEGIN readln(k); i:=1; while f(i)<k do i:=i+1; write(i); END.</pre>	<pre>int f(int n) { int i,s; s = 0; for(i=1 ; i<=n ; i++) s = s + n; return s; } void main() { int k, i; scanf("%d",&k); i = 1; while (f(i)<k) i++; printf("%d", i); }</pre>
Бейсик	Алгоритмический язык
<pre>DIM K, I AS INTEGER INPUT K I = 1 WHILE F(I) < K I = I + 1 WEND PRINT I FUNCTION F(N) DIM I,S AS INTEGER S = 0 FOR I = 1 TO N S = S + N NEXT I F = S END FUNCTION</pre>	<pre>алг нач цел k, i ввод k i := 1 нц пока F(i) < k i := i + 1 кц вывод i кон алг цел F(цел n) цел i, s нач s := 0; нц для i от 1 до n s := s + n кц знач := s кон</pre>
Python	
<pre>def f(n): s = 0 for i in range(1,n+1): s = s + n return s k = int(input()) i = 1 while f(i)<k: i++ print(i)</pre>	

Ответ: _____ .

22. Исполнитель Калькулятор преобразует число, записанное на экране.

У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1,
2. Прибавить 2,
3. Умножить на 2.

Первая из них увеличивает число на экране на 1, вторая увеличивает его на 2, третья умножает его на 2.

Программа для исполнителя Калькулятор — это последовательность команд.

Сколько существует таких программ, которые исходное число 3 преобразуют в число 12, и при этом траектория вычислений программы не содержит число 6?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 123 при исходном числе 1 траектория будет состоять из чисел 2, 4, 8.

Ответ: _____.

23. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_8, x_9$, которые удовлетворяют всем перечисленным ниже условиям?

$$x_1 \vee x_2 \wedge x_3 = 1$$

$$x_2 \vee x_3 \wedge x_4 = 1$$

...

$$x_6 \vee x_7 \wedge x_8 = 1$$

$$x_7 \vee x_8 \wedge x_9 = 1$$

Ответ: _____.

Часть 2

Для записи ответов на задания этой части (24–27) используйте бланк ответов № 2. Запишите сначала номер задания (24, 25 и т.д.), а затем полное решение. Ответы записывайте четко и разборчиво.

24. Требовалось написать программу, которая считывает с клавиатуры n натуральных чисел (например, $n = 4$), каждое из которых не больше 1000, и выводит на экран сумму тех из них, которые не кратны 3. Если таких чисел нет, программа должна вывести на экран «NO».

Программист торопился и написал программу неправильно.

Паскаль	Си
<pre>const n = 4; var i, x : integer; sum, k : integer; begin k := 1; sum := 0; for i := 1 to n do begin read(x); if x mod 3 <> 0 then begin k := k + 1; sum := sum + x end end; if k > 0 then writeln(k) else writeln('NO') end.</pre>	<pre>#include <stdio.h> #define n 4 void main(void) { int i, x; int sum, k; k = 1; sum = 0; for (i = 1; i <= n; i++) { scanf("%d", &x); if (x % 3 != 0) { k++; sum = sum + x; } } if (k > 0) printf("%d\n", k); else printf("NO\n"); }</pre>

Бейсик	Алгоритмический язык
<pre> CONST n = 4 k = 1 sum = 0 FOR I = 1 TO n INPUT x IF x mod 3 <> 0 THEN k = k + 1 sum = sum + x END IF NEXT I IF k > 0 THEN PRINT k ELSE PRINT "NO" END IF </pre>	<pre> алг нач цел n = 4 цел i, x цел sum, k k := 1 sum := 0 нц для i от 1 до n ввод x если mod(x, 3) <> 0 то k := k + 1 sum := sum + x все кц если k > 0 то вывод k иначе вывод "NO" все кон </pre>
Python	
<pre> N=4 k = 1 sum = 0 for i in range(1,n+1): x = int(input()) if x%3 != 0 : k++ sum = sum + x if k > 0 : print(k); else : print("NO") </pre>	

Последовательно выполните следующее.

1. Укажите, что выведет на экран программа при вводе последовательности чисел 1 2 3 4.
2. Приведите пример такой последовательности из n допустимых чисел, при котором программа работает верно.
3. Исправьте все ошибки в программе. Для этого для каждой ошибки приведите строку, которая написана неверно, и строку, на которую ее нужно заменить, чтобы программа работала верно.

25. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит сумму элементов самой длинной возрастающей последовательности подряд идущих элементов массива. Если таких последовательностей несколько, вывести сумму элементов самой первой такой последовательности.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre> const N=30; var a: array [1..N] of integer; i, j, k, max, sum: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre> N=30 DIM A(N) AS INTEGER DIM I, J, K, MAX, SUM AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Естественный язык
<pre> #include <stdio.h> #define N 30 void main(void) {int a[N]; int i, j, k, max, sum; for (i=0; i<N; i++) scanf("% d", &a[i]); ... }</pre>	<p>Объявляем массив А из 30 элементов.</p> <p>Объявляем целочисленные переменные I, J, K, MAX, SUM.</p> <p>В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й.</p> <p>...</p>
Python	
<pre> N = 40 i = None j = None k = None a = [int(input()) for i in range(N)] ...</pre>	

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учетом синтаксиса и особенностей используемого вами языка программирования.

26. Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один** камень или увеличить количество камней в куче **в два раза**. Например, имея кучу из 6 камней, за один ход можно получить кучу из 7 или 12 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 18. Если при этом в куче оказалось не более 26 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 17 камней и Паша удвоит количество камней в куче, то игра закончится и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 17$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания.

1. а) При каких значениях числа S Паша может выиграть в один ход?

Укажите все такие значения и соответствующие ходы Паши.

б) У кого из игроков есть выигрышная стратегия при $S = 16, 15, 14$?

Опишите выигрышные стратегии для этих случаев.

2. У кого из игроков есть выигрышная стратегия при $S = 8, 7$? Опишите соответствующие выигрышные стратегии.

3. У кого из игроков есть выигрышная стратегия при $S = 6$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах — количество камней в позиции.

27. Вам предлагается два задания с похожими условиями: задание А и задание Б. Вы можете решать оба задания или одно из них по своему выбору. Задание Б более сложное, его решение оценивается выше. Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б.

Задание А. Имеется набор данных, состоящий из 6 пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 7 и при этом была минимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Напишите программу для решения этой задачи. В этом варианте задания оценивается только правильность программы, время работы и размер использованной памяти не имеют значения.

Максимальная оценка за правильную программу — 2 балла.

Задание Б. Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 7 и при этом была минимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Напишите программу для решения этой задачи.

Постарайтесь сделать программу эффективной по времени и по используемой памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству пар чисел N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, — 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но не эффективную по памяти, — 3 балла.

Как в варианте А, так и в варианте Б программа должна напечатать одно число — минимально возможную сумму, соответствующую условиям задачи (или 0, если такую сумму получить нельзя).

Перед текстом программы кратко опишите ваш алгоритм решения, укажите использованный язык программирования и его версию (например, Free Pascal 2.6.4).

Входные данные

Для варианта А на вход программе подаётся 6 строк, каждая из которых содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта А:

```
1 7
7 14
28 3
5 9
4 3
2 2
```

Для варианта Б на вход программе в первой строке подаётся количество пар N ($1 \leq N \leq 100\,000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта Б:

```
6
1 7
7 14
28 3
5 9
4 3
2 2
```

Пример выходных данных для приведённых выше примеров входных данных:

```
22
```

ОТВЕТЫ К КОНТРОЛЬНЫМ ВАРИАНТАМ ЭКЗАМЕНАЦИОННОЙ РАБОТЫ

Вариант 1

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
1	15	11	64	21	19
2	cabd	12	GCD A	22	36
3	11	13	160	23	21
4	42	14	8877		
5	11	15	32		
6	1110	16	5		
7	24	17	23		
8	50	18	14		
9	192	19	1		
10	405	20	135		

24

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

<p>Решение использует запись программы на Паскале. Допускается использование программы на трех других языках.</p> <p>1. Программа выведет число: 3.</p> <p>2. Пример последовательности, для которой программа работает правильно: 2 2 3 2. Замечание для проверяющего. Вместо минимального нечетного программа находит максимальное нечетное. Значит, верным будет любая последовательность 4-х натуральных чисел, которая либо не содержит нечетных элементов, либо этот элемент единственный, либо все нечетные элементы равны друг другу.</p> <p>3. В программе есть две ошибки.</p> <p>Первая ошибка: неверная инициализация переменной min. Строка с ошибкой: min := 0; Верное исправление: min:= 1001; Допускается любое число, больше и равное 1000.</p> <p>Вторая ошибка: неверная проверка на минимальность. Строка с ошибкой: if min < x then Верное исправление: if min > x then</p>	
Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <p>1) указать, что выведет программа при конкретной входной последовательности;</p> <p>2) указать пример последовательности, при которой программа работает правильно;</p>	

<p>3) исправить первую ошибку; 4) исправить вторую ошибку.</p> <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p> <p>Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</p>	
<p>Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: а) выполнены три из четырех необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки</p>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырех</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2 или 3 балла</p>	0
<p><i>Максимальный балл</i></p>	3

25

Содержание верного ответа и указания по оцениванию
(допускаются иные формулировки ответа, не искажающие его смысла)

На языке Паскаль

```
k := 0;
for i := 1 to N do
  if (a[i] mod 10 + a[i] div 10) mod 3 <> 0 then
    k := k + 1;
writeln(k);
```

На алгоритмическом языке

```
k := 0
нц для i от 1 до N
  если mod(mod(a[i],10)+div(a[i],10)), 3 <>0
  то
    k := k + 1
  все
кц
вывод k
```

На языке Бейсик

```
K = 0
FOR I = 1 TO N
IF (A(I) MOD 10+A(I) \ 10)MOD 3 <>0 THEN
K = K + 1
ENDIF
NEXT I
PRINT K
```

На языке СИ

```
k=0;
for (i=0; i<N; i++)
    if a[i]%10+a[i]/10 %3 != 0
        k++;
printf("%d", k);
```

На языке Python

```
k=0
a = [int(input())]
for i in range(0, N):
    if a[i]%10+a[i]/10 %3 !=0 :
        k++
print(k)
```

Указания по оцениванию

Баллы

Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы

2

В любом варианте решения может присутствовать не более одной ошибки из числа следующих.

- Не инициализируется или неверно инициализируется переменная k (например, присваивается начальное значение, не равное 0).
- Неверно осуществляется выделение цифр числа.
- На делимость на 3 проверяется не сумма цифр числа, а что-то другое.
- Неверно осуществляется проверка неделимости на три.
- В сложном условии вместо логической операции «И» используется логическая операция «ИЛИ».
- Неверно осуществляется накопление количества элементов в цикле (например, $k := 1$).
- Отсутствует вывод ответа.
- Используется переменная, не объявленная в разделе описания переменных.
- Не указано или неверно указано условие завершения цикла.
- Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно, или отсутствует цикл.
- Неверно расставлены операторные скобки

1

Ошибок, перечисленных выше, две или больше, или алгоритм сформулирован неверно

0

Максимальный балл

2

26

Содержание верного ответа и указания к оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание 1. В начальных позициях (4, 28), (6, 27) выигрышная стратегия есть у Вити. При начальной позиции (4, 28) после первого хода Паши может получиться одна из следующих четырех позиций: (5, 28), (8, 28), (4, 56), (4, 29). Каждая из этих позиций содержит менее 61 камня. При этом из

любой из этих позиций Витя может получить позицию, содержащую не менее 61 камня, удвоив количество камней во второй куче. Для позиции (6, 27) после первого хода Паши может получиться одна из следующих четырех позиций: (7, 27), (12, 27), (6, 28), (6, 54). Каждая из этих позиций содержит менее 61 камня. При этом из любой из этих позиций Витя может получить позицию, содержащую не менее 61 камня, удвоив количество камней во второй куче. Таким образом, Витя при любом ходе Паши выигрывает своим первым ходом.

Задание 2. В начальных позициях (4, 27), (5, 27) и (6, 26) выигрышная стратегия есть у Паши. При начальной позиции (4, 27) он должен первым ходом получить позицию (4, 28), из начальных позиций (5, 27) и (6, 26) Паша после первого хода должен получить позицию (6, 27). Позиции (4, 28) и (6, 27) рассмотрены при разборе задания 1. В этих позициях выигрышная стратегия есть у игрока, который будет ходить вторым (теперь это Паша). Эта стратегия описана при разборе задания 1. Таким образом, Паша при любой игре Вити выигрывает своим вторым ходом.

Задание 3. В начальной позиции (5, 26) выигрышная стратегия есть у Вити. После первого хода Паши может возникнуть одна из четырех позиций: (6, 26), (5, 27), (10, 26) и (5, 52). В позициях (10, 26) и (5, 52) Витя может выиграть одним ходом, удвоив количество камней во второй куче. Позиции (6, 26) и (5, 27) были рассмотрены при разборе задания 2. В этих позициях у игрока, который должен сделать ход (теперь это Витя), есть выигрышная стратегия. Эта стратегия описана при разборе задания 2. Таким образом, в зависимости от игры Паши Витя выигрывает на первом или на втором ходу.

В таблице изображено дерево возможных партий при описанной стратегии Вити. Заключительные позиции (в них выигрывает Витя) выделены жирным шрифтом.

Исходное положение	Положения после очередных ходов			
	1-й ход Паши (разобраны все ходы, указана полученная позиция)	1-й ход Вити (только ход по стратегии, указана полученная позиция)	2-й ход Паши (разобраны все ходы, указана полученная позиция)	2-й ход Вити (только ход по стратегии, указана полученная позиция)
(5, 26) Всего: 31	(5, 26+1) = (5, 27) Всего: 32	(5+1, 27) = (6, 27) Всего: 33	(6+1, 27) = (7, 27) Всего: 34	(7, 27*2) = (7, 54) Всего: 61
			(6, 27+1) = (6, 28) Всего: 34	(6, 28*2) = (6, 56) Всего: 62
			(6*2, 27) = (12, 27) Всего: 39	(12, 27*2) = (12, 54) Всего: 66
			(6, 27*2) = (6, 54) Всего: 60	(6, 54*2) = (6, 108) Всего: 114
	(5+1, 26) = (6, 26) Всего: 32	(6, 26+1) = (6, 27) Всего: 33	(6+1, 27) = (7, 27) Всего: 34	(7, 27*2) = (7, 54) Всего: 61
			(6, 27+1) = (6, 28) Всего: 34	(6, 28*2) = (6, 56) Всего: 62
			(6*2, 27) = (12, 27) Всего: 39	(12, 27*2) = (12, 54) Всего: 66
			(6, 27*2) = (6, 54) Всего: 60	(6, 54*2) = (6, 108) Всего: 114
	(5*2, 26) = (10, 26) Всего: 36	(10, 26*2) = (10, 52) Всего: 62		
	(5, 26*2) = (5, 52) Всего: 57	(5, 52*2) = (5, 104) Всего: 109		

Примечание. Дерево всех партий может быть также изображено в виде ориентированного графа. Вершины дерева, соответствующие одной и той же позиции, на рисунке могут быть «склеены». Важ-

но, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

Примечание. В некоторых позициях у Вити есть и другой способ выигрыша: например, в позиции (6, 54) можно добавить один камень в любую кучу. То, что это не указано, не является ошибкой. Экзаменуемый не должен указывать все возможные выигрышные стратегии.

27

Основные элементы правильного ответа:

Программа читает все входные данные только один раз, запоминая в целочисленном массиве из 9 элементов, сколько раз встречается соответствующая цифра во входных данных.

После этого проверяется, что все ненулевые элементы этого массива равны трем. Если это условие не выполняется или все элементы массива равны нулю, то выводится «NO».

В противном случае три раза подряд выводится упорядоченная по возрастанию группа цифр, состоящая из всех ненулевых цифр исходной строки.

Пример возможного решения (на C++):

```
#include <iostream>
using namespace std;

int main ()
{
    int num[10]; //из соображений наглядности элемент num[0] не задействован
    char c;
    for (int i=1; i<=9; i++) num[i] = 0; //инициализация
    cin >> c;
    while (c != '0')
    {
        if ((c>='1') && (c<='9')) num[c-'0']++; //подсчет цифр
        cin >> c;
    }

    bool yes =false;
    for (int i=1; i<=9; i++)
    //проверка, есть ли в строке хотя бы одна цифра
    if (num[i]> 0){
        yes = true;
        break;
    }
    if (yes)
    {
        for (int i=1; i<=9; i++)
        //проверка, кратно ли трем количество вхождений каждой встреченной цифры
        if ((num[i]> 0) && (num[i]%3!= 0)){
            yes = false;
            break;
        }
    }
    if (yes) //вывод результата
    {
        for (int j=1; j<=3; j++)
            for (int i=1; i<=9; i++)
                for (int k=1; k<= num[i] / 3; k++) cout << i;
    }
    else cout << "NO";
}
```

```

cout << '\n';
return 0;
}

```

Пример возможного решения (на классическом Паскале):

```

program c4 (input, output);
var  num: array ['1'.. '9'] of integer;
     i,c: char;
     j,k: integer;
     yes: boolean;
begin
for i:= '1' to '9' do num[i] := 0; {инициализация}
read (c);
while c <> '0' do
begin
    if (c>='1') and (c<='9') then num[c] := num[c]+1; {подсчет цифр}
    read (c)
end;
yes := false;
i:= '1';
while (not yes) and (i<='9') do
begin
    {проверка, есть ли в строке хотя бы одна цифра}
    if num[i] > 0 then yes := true;
    i:= succ(i)
end;
if yes then
begin
    i:= '1';
    while yes and (i<='9') do
{проверка, кратно ли трем количество вхождений каждой встреченной цифры}
    begin
        if (num[i]>0) and (num[i] mod 3 <>0) then yes := false;
        i:= succ(i)
    end
end;
if yes then {вывод результата}
begin
    for j:=1 to 3 do
        for i:= '1' to '9' do
            for k:=1 to num[i] div 3 do write(i) end
        else write ('NO');
    writeln;
end.

```

Пример возможного решения (на Бейсике, компилятор FreeBASIC):

```
Dim num(9) As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim yes As Integer
Dim c As String
For i=1 To 9
  Num (i) = 0 'инициализация
Next i
Input c
Do While c<>"0"
  'подсчет цифр
  if (c>"0") And (c<="9") Then Num(Asc(c)-Asc("0"))=
= Num(Asc(c)- Asc("0"))+1
  Input c
Loop

yes = 0
For i=1 To 9
'проверка, есть ли в строке хотя бы одна цифра
  If num(i) > 0 Then
    yes = 1
    Exit For
  End If
Next i

If yes = 1 Then
  For i=1 To 9
    'проверка, кратно ли трем количество вхождений каждой встреченной цифры
    If (num(i)>0) And (num(i) Mod 3<> 0) Then
      yes = 0
      Exit For
    End If
  Next i
End If
c=""
If yes = 1 Then 'вывод результата
  For j=1 To 3
    For i=1 To 9
      For k=1 To num(i)/3
        c=c+Chr$(i+Asc("0"))
      Next k
    Next i
  Next j
  Print c
Else
  Print "NO"
End If
```

Вариант 2

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
1	17	11	15	21	15
2	dbac	12	CGDA	22	25
3	9	13	140	23	50
4	84	14	23		
5	010	15	19		
6	9896	16	2014		
7	27	17	18		
8	51	18	11		
9	9	19	100		
10	65	20	224		

24

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

<p>Решение использует запись программы на Паскале. Допускается использование программы на трех других языках.</p> <p>1. Программа выведет число: 4.</p> <p>2. Пример последовательности, для которой программа работает правильно: 1 1 1 2. Замечание для проверяющего. В конце программы выводимое на экран значение переменной k будет на 1 больше, чем количество введенных чисел, не кратных трем. То есть верной будет любая последовательность, у которой сумма чисел, не кратных трем, равна количеству этих чисел +1. Так как числа натуральные, это возможно, только если в последовательности есть числа, не кратные трем, одно из них = 2, остальные = 1.</p> <p>3. В программе есть две ошибки. Первая ошибка: неверная инициализация переменной k. Строка с ошибкой: <code>k := 1;</code> Верное исправление: <code>k := 0;</code> Вторая ошибка: неверный вывод на экран (выводится не та переменная). Строка с ошибкой: <code>writeln(k);</code> Верное исправление: <code>writeln(sum);</code></p>	
Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <ol style="list-style-type: none"> 1) указать, что выведет программа при конкретной входной последовательности; 2) указать пример последовательности, при которой программа работает правильно; 3) исправить первую ошибку; 4) исправить вторую ошибку. <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы</p>	

Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: а) выполнены три из четырех необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырех	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

```

k:=1; { Число элементов в текущей последовательности.
      Изначально считаем, что последовательность состоит
      из одного (первого) элемента }
sum:=a[1]; { Сумма элементов текущей последовательности }
max:=0; { Наибольшая длина возрастающей последовательности }
for i:=2 to n do { Перебираем все пары соседних элементов }
  if a[i]>a[i-1] then { Если последовательность продолжает возрастать }
  begin
    sum:=sum+a[i]; { Увеличиваем ее сумму элементов }
    k:=k+1        { Увеличиваем счетчик длины последовательности }
  end
  else { Если последовательность прекратила возрастать }
  begin
    if k>max then { Проверяем, не является ли закончившаяся последовательность
                  длиннее текущего максимума }
    begin { Если да: }
      max:=k; { Меняем текущий максимум }
      j:=sum  { Запоминаем сумму его элементов }
    end;
    k:=1;    { Устанавливаем длину новой последовательности равной 1 }
    sum:=a[i] { Сумма элементов этой последовательности (пока) }
  end;
if k>max then { На случай, если самая длинная последовательность находится
              в конце массива, проверим еще раз текущий максимум }
begin
  max:=k;
  j:=sum
end;
write(j); { Вывод ответа на экран }
end.

```

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Паша может выиграть в один ход, если $S = 17$ или $S = 9, 10, 11, 12, 13$. При $S = 17$ первым ходом нужно добавить в кучу один камень, при остальных указанных значениях S нужно удвоить количество камней.

б) При $S = 16, 15$ или 14 удваивать количество камней не имеет смысла, так как после такого хода выигрывает противник. Поэтому можно считать, что единственный возможный ход — это добавление в кучу одного камня.

При $S = 16$ после такого хода Паши в куче станет 17 камней. В этой позиции ходящий (т.е. Валя) выигрывает (см. п. а)), значит, при $S = 16$ Паша (игрок, который должен ходить первым) проигрывает. Выигрышная стратегия есть у Вали.

При $S = 15$, после того как Паша своим первым ходом добавит один камень, в куче станет 16 камней. В этой позиции ходящий (т.е. Валя) проигрывает (см. выше), следовательно, при $S = 15$ Паша (игрок, который должен ходить первым) выигрывает. Выигрышная стратегия есть у Паши.

При $S = 14$ выигрышная стратегия есть у Вали. Действительно, если Паша первым ходом удваивает количество камней, то игра сразу заканчивается выигрышем Вали. Если Паша добавляет один камень, то в куче становится 15 камней. Как мы уже знаем, в этой позиции игрок, который должен ходить (т.е. Валя), выигрывает.

Во всех случаях выигрыш достигается тем, что при своём ходе игрок, имеющий выигрышную стратегию, должен добавить в кучу один камень.

2. При $S = 8$ или 7 выигрышная стратегия есть у Паши. Она состоит в том, чтобы удвоить количество камней в куче и получить кучу, в которой будет соответственно 16 или 14 камней. В обоих случаях игрок, который будет делать ход (теперь это Валя), проигрывает (п. 1б):

3. При $S = 6$ выигрышная стратегия есть у Вали. После первого хода Паши в куче может стать либо 7, либо 12 камней. В обеих этих позициях выигрывает игрок, который будет делать ход (теперь это Валя). Случай $S = 7$ рассмотрен в п. 2, случай $S = 12$ рассмотрен в п. 1а).

В таблице изображено дерево возможных партий при описанной стратегии Вали. Заключительные позиции (в них выигрывает Валя) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов						
И.п.	1-й ход Паши (все ходы)	1-й ход Вали (только ход по стратегии)	2-й ход Паши (все ходы)	2-й ход Вали (только ход по стратегии)	3-й ход Паши (все ходы)	3-й ход Вали (только ход по стратегии)
6	$6+1=7$	$7*2=14$	$14+1=15$	$15+1=16$	$16+1=17$	<u>$17+1=18$</u>
					<u>$16*2=32$</u>	
			<u>$14*2=28$</u>			
	$6*2=12$	<u>$12*2=24$</u>				

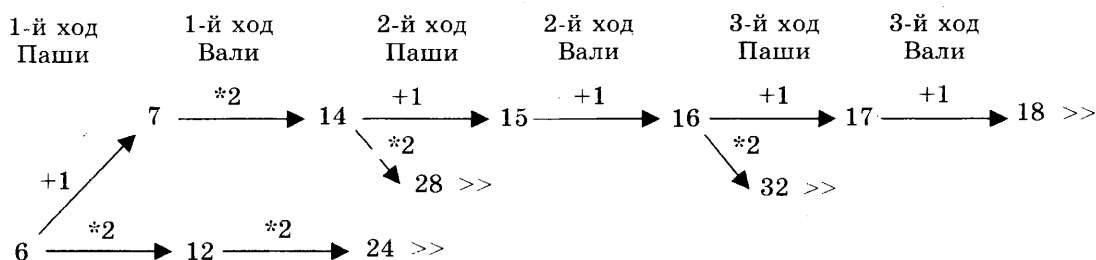


Рис.1. Дерево всех партий, возможных при Валиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание Б.

Чтобы получить минимально возможную сумму, будем брать из каждой пары наименьшее число. Если полученная при этом сумма будет делиться на 7, её необходимо увеличить. Для этого достаточно в одной из пар, где числа имеют разные остатки при делении на 7, заменить ранее выбранное число на другое число из той же пары. При этом разница между числами в паре должна быть минимально возможной. Если во всех парах оба числа имеют одинаковый остаток при делении на 7, получить нужную сумму невозможно.

Задание А.

В цикле перебираются все возможные суммы и среди них ищется удовлетворяющая условию.

Пример правильной и эффективной программы для задания Б на языке Паскаль

```

const
  aMax = 10000; {наибольшее возможное число в исходных данных}

var
  N: longint; {количество пар}
  a, b: longint; {пара чисел}
  Max: longint; {максимум в паре}
  Min: longint; {минимум в паре}
  s: longint; {сумма выбранных чисел}
  D_min: longint; {минимальная разница Max-Min не кратная 7}
  i: longint;

begin
  s := 0;
  D_min := aMax + 1;
  readln(N);
  for i := 1 to N do begin
    readln(a, b);
    if a>b then begin Max:=a; Min:=b end
      else begin Max:=b; Min:=a end;
    s := s + Min;
    if ((Max - Min) mod 7 > 0) and (Max - Min < D_min)
      then D_min := Max - Min
  end;
  if s mod 7 = 0 then begin
    if D_min > aMax then s := 0
      else s := s + D_min
  end;
  writeln(s)
end.

```

Пример решения задания А на языке Паскаль

```
var
  a: array[1..6, 1..2] of longint;
  i1, i2, i3, i4, i5, i6: longint;
  s, sMin: longint;
begin
  for i1:= 1 to 6 do readln(a[i1,1], a[i1,2]);
  sMin := 100000;
  for i1:=1 to 2 do
    for i2:=1 to 2 do
      for i3:=1 to 2 do
        for i4:=1 to 2 do
          for i5:=1 to 2 do
            for i6:=1 to 2 do begin
              s:=a[1,i1]+a[2,i2]+a[3,i3]+a[4,i4]+a[5,i5]+a[6,i6];
              if (s mod 7 <> 0) and (s < sMin) then sMin := s
            end;
          if sMin = 100000 then sMin :=0;
          writeln(sMin)
        end.
end.
```


Справочное издание

**Крылов Сергей Сергеевич
Ушаков Денис Михайлович**

ЕГЭ ИНФОРМАТИКА ТРЕНАЖЁР

Издательство «**ЭКЗАМЕН**»

Гигиенический сертификат
№ РОСС RU.АД44.Н02841 от 30.06.2017 г.

Главный редактор *Л. Д. Лаппо*
Редактор *Г. А. Лонцова*
Технический редактор *Л. В. Павлова*
Корректоры *Н. Е. Жданова, Л. В. Краденых*
Дизайн обложки *Л. В. Демьянова*
Компьютерная верстка *А. В. Толокевич, М. А. Серова*

107045, Москва, Луков пер., д. 8.
www.examen.biz

E-mail: по общим вопросам: info@examen.biz;
по вопросам реализации: sale@examen.biz
тел./факс 8 (495) 641-00-30 (многоканальный)

Общероссийский классификатор продукции
ОК 005-93, том 2; 953005 — книги, брошюры, литература учебная

Отпечатано в соответствии с предоставленными материалами
в ООО «Красногорская типография», 143405, Московская обл., г. Красногорск,
Коммунальный квартал, д.2. www.ktprint.ru

По вопросам реализации обращаться по тел.:
8 (495) 641-00-30 (многоканальный).