

**Сборник тренировочных материалов для подготовки  
к государственной итоговой аттестации по ИНФОРМАТИКЕ и ИКТ  
для слепых и поздноослепших обучающихся  
по образовательным программам  
СРЕДНЕГО общего образования**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Тренировочные материалы предназначены для подготовки к единому государственному экзамену и государственному выпускному экзамену (в письменной форме) по информатике.

Тренировочные материалы состоят из пяти разделов, в которых задания сгруппированы следующим образом:

- раздел 1 – информация и её кодирование;
- раздел 2 – системы счисления, моделирование и компьютерный эксперимент, основы логики;
- раздел 3 – архитектура компьютеров и компьютерных сетей, обработка числовой информации, базы данных, поиск информации в сети;
- раздел 4 – алгоритмы, элементы теории алгоритмов;
- раздел 5 – программирование.

**Раздел 1** содержит 17 заданий. Верное выполнение каждого из заданий 1–17 оценивается в 1 балл. Задания 1–17 могут включаться как в экзаменационные материалы ГВЭ-11, так и в КИМ ЕГЭ.

**Раздел 2** содержит 26 заданий. За верное выполнение каждого из заданий 1–26 выставляется по 1 баллу. Задания 1–23 могут включаться как в экзаменационные материалы ГВЭ-11, так и в КИМ ЕГЭ, а задания 24–26 – только в КИМ ЕГЭ.

**Раздел 3** содержит 16 заданий. Верное выполнение каждого из заданий 1–16 оценивается в 1 балл. Задания 1–16 могут включаться как в экзаменационные материалы ГВЭ-11, так и в КИМ ЕГЭ.

**Раздел 4** содержит 32 задания. За верное выполнение каждого из заданий 1–26 выставляется по 1 баллу. Ответ на задания 27–29 оценивается максимально в 2 балла, а ответ на задание 30–32 – в 3 балла в соответствии с критериями оценивания. Задания 1–29 могут включаться как в экзаменационные материалы ГВЭ-11, так и в КИМ ЕГЭ, задания 30–32 используются только в КИМ ЕГЭ.

**Раздел 5** содержит 14 заданий. За верное выполнение каждого из заданий 1–10 выставляется по 1 баллу. Ответ на задания 11 и 12 оценивается максимально в 3 балла, а на задания 13 и 14 – в 4 балла в соответствии с критериями оценивания. Задания 1–10 могут включаться как в экзаменационные материалы ГВЭ-11, так и в КИМ ЕГЭ, задания 11–14 используются только в КИМ ЕГЭ.

**РАЗДЕЛ 1**

*Ответом к заданиям 1–17 является одна цифра или число. Ответ к заданиям 1–17 запишите в отведённом месте работы.*

**Информация и её кодирование****1**

В некоторой информационной системе информация кодируется двоичными шестиразрядными словами. При передаче данных возможны их искажения, поэтому перед передачей в конец каждого слова добавляется седьмой (контрольный) разряд, таким образом, чтобы сумма разрядов нового слова, считая контрольный, была чётной. Например, к слову 110011 справа будет добавлен 0, а к слову 101100 – 1.

После приёма слова производится его обработка. При этом проверяется сумма его разрядов, включая контрольный. Если она нечётна, это означает, что при передаче этого слова произошёл сбой, и оно автоматически заменяется на зарезервированное слово 0000000. Если она чётна, это означает, что сбоя не было или сбоев было больше одного. В этом случае слово не изменяется.

Исходное сообщение

1010101 0101011 0111010

было принято в виде

1010111 0101011 0111001.

Как будет выглядеть принятое сообщение после обработки?

- 1) 0000000 0101011 0111001
- 2) 1010111 0000000 0000000
- 3) 0000000 0101011 0000000
- 4) 1010111 0000000 0111001

Ответ:

2

Для передачи данных по каналу связи используется 5-битовый код. Сообщение содержит только буквы А, Б и В, которые кодируются следующими кодовыми словами:

А – 11011; Б – 01110; В – 10000.

При передаче возможны помехи. Однако некоторые ошибки можно попытаться исправить. Любые два из этих трёх кодовых слов отличаются друг от друга не менее чем в трёх позициях. Поэтому если при передаче слова произошла ошибка не более чем в одной позиции, то можно сделать обоснованное предположение о том, какая буква передавалась. (Говорят, что «код исправляет одну ошибку».) Например, если получено кодовое слово 11110, считается, что передавалась буква Б. (Отличие от кодового слова для Б только в одной позиции, для остальных кодовых слов отличий больше.) Если принятое кодовое слово отличается от кодовых слов для букв А, Б, В более чем в одной позиции, то считается, что произошла ошибка, которую нельзя исправить. Такая ошибка обозначается 'х'.

Получено сообщение 10010 11010 00110 01011. Декодируйте это сообщение – выберите правильный вариант.

- 1) хххх                      2) ВАБА                      3) ВхБх                      4) ВАБх

Ответ:

3

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 приведённых 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

Ответ:

**4** Некоторый алфавит содержит 7 различных букв. Сколько 3-буквенных слов можно составить из букв данного алфавита (буквы в слове могут повторяться)?

Ответ: \_\_\_\_\_.

**5** Все 3-буквенные слова, составленные из букв Г, Е, П, А, Р, Д, записаны в алфавитном порядке и перенумерованы, начиная с 1.

Ниже приведено начало списка.

1. ААА
2. ААГ
3. ААД
4. ААЕ
5. ААП
6. ААР
7. АГА

Под каким номером в списке идёт первое слово, которое начинается с буквы Г?

Ответ: \_\_\_\_\_.

**6** Все 3-буквенные слова, составленные из букв У, Ч, Е, Н, И, К, записаны в алфавитном порядке и перенумерованы, начиная с 1.

Ниже приведено начало списка.

1. ЕЕЕ
2. ЕЕИ
3. ЕЕК
4. ЕЕН
5. ЕЕУ
6. ЕЕЧ
7. ЕИЕ

Под каким номером в списке идёт первое слово, которое начинается с буквы К?

Ответ: \_\_\_\_\_.

**7** Вася составляет 5-буквенные слова, в которых есть только буквы С, Л, О, Н, причём буква С 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?

Ответ: \_\_\_\_\_.

**8** Некоторый алфавит содержит четыре различные буквы. Сколько пятибуквенных слов можно составить из букв данного алфавита (буквы в слове могут повторяться)?

Ответ: \_\_\_\_\_.

**9** Музыкальный фрагмент был записан в формате стерео (двухканальная запись), оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла – 48 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате моно и оцифрован с разрешением в 1,5 раза выше и частотой дискретизации в 3 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: \_\_\_\_\_.

**10** 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.

Ответ: \_\_\_\_\_.

**11** Музыкальный фрагмент был записан в формате стерео (двухканальная запись), затем оцифрован и сохранён в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 18 секунд. Затем тот же музыкальный фрагмент был повторно записан в формате моно и оцифрован с разрешением в 2 раза выше 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 5 раз 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.

Ответ: \_\_\_\_\_.

**12**

Музыкальный фрагмент был записан в формате моно, оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла – 30 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: \_\_\_\_\_.

**13**

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 8-символьного набора: А, В, С, D, E, F, G, H. В базе данных для хранения пароля отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит.

Определите объём памяти (в байтах), необходимый для хранения паролей 20 пользователей. В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.

**14**

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 8-символьного набора: А, В, С, D, E, F, G, H. 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 24 байт на одного пользователя.

Определите объём памяти (в байтах), необходимый для хранения сведений о 30 пользователях. В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.

15

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 8 символов и содержащий только символы из 26-символьного набора прописных латинских букв. 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 выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 30 пользователях потребовалось 600 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.

16

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 6 символов и содержащий только символы из 26-символьного набора прописных латинских букв. 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 выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 20 пользователях потребовалось 400 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.

17

По каналу связи передаются сообщения, содержащие только 4 буквы: А, Б, В, Г; для передачи используется двоичный код, удовлетворяющий условию Фано. Для букв Б, В, Г используются такие кодовые слова: Б – 0; В – 100; Г – 110.

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

*Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: \_\_\_\_\_.

**Ответы к заданиям**

№ задания	Ответ
1	1
2	2
3	1
4	343
5	37
6	73
7	405
8	1024
9	12
10	10
11	180
12	80
13	120
14	900
15	15
16	16
17	101



**РАЗДЕЛ 2**

*Ответом к заданиям 1–26 является одна цифра, число или последовательность латинских букв. Ответы к заданиям 1–26 запишите в отведённом месте работы.*

**Системы счисления**

**1** Сколько единиц в двоичной записи десятичного числа 513?

Ответ: \_\_\_\_\_.

**2** Сколько единиц в двоичной записи десятичного числа 1023?

Ответ: \_\_\_\_\_.

**3** Сколько единиц в двоичной записи восьмеричного числа  $1731_8$ ?

Ответ: \_\_\_\_\_.

**4** Сколько единиц в двоичной записи восьмеричного числа  $1707_8$ ?

Ответ: \_\_\_\_\_.

**5** Сколько единиц в двоичной записи шестнадцатеричного числа  $11F0_{16}$ ?

Ответ: \_\_\_\_\_.

**6** Сколько единиц в двоичной записи шестнадцатеричного числа  $E0A1_{16}$ ?

Ответ: \_\_\_\_\_.

**7**

Сколько значащих нулей содержится в десятичной записи значения выражения:

$$100^{10} - 10^6 + 100?$$

Ответ: \_\_\_\_\_.

**8**

Сколько единиц содержится в двоичной записи значения выражения:

$$4^{16} + 2^{36} - 16?$$

Ответ: \_\_\_\_\_.

**9**

Сколько единиц содержится в двоичной записи значения выражения:

$$4^{2014} + 2^{2015} - 9?$$

Ответ: \_\_\_\_\_.

**10**

Значение арифметического выражения:  $9^{12} + 3^7 - 27$  – записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Ответ: \_\_\_\_\_.

**11**

Значение арифметического выражения:  $9^{30} + 3^{90} - 9$  – записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Ответ: \_\_\_\_\_.

**Моделирование и компьютерный эксперимент**

**12**

Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		2				12
B	2		2	4		
C		2		3		
D		4	3		3	5
E				3		1
F	12			5	1	

Определите длину кратчайшего пути между пунктами А и F (при условии, что передвигаться можно только по указанным в таблице дорогам.)

Ответ: \_\_\_\_\_.

**13**

Между населёнными пунктами А, Б, В, Г, Д, Е и К построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	Б	В	Г	Д	Е	К
A		8					
Б	8		3	5	8		
В		3			4		
Г		5			1		3
Д		8	4	1		2	1
Е					2		1
К				3	1	1	

Определите длину кратчайшего пути между пунктами А и К (при условии, что передвигаться можно только по построенным дорогам).

Ответ: \_\_\_\_\_.

14

Между населёнными пунктами А, Б, В, Г, Д, Е и К построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	А	Б	В	Г	Д	Е	К
А		5					
Б	5		3	5	8		
В		3			1		
Г		5			1		3
Д		8	1	1		2	5
Е					2		1
К				3	5	1	

Определите длину кратчайшего пути между пунктами А и К (при условии, что передвигаться можно только по построенным дорогам).

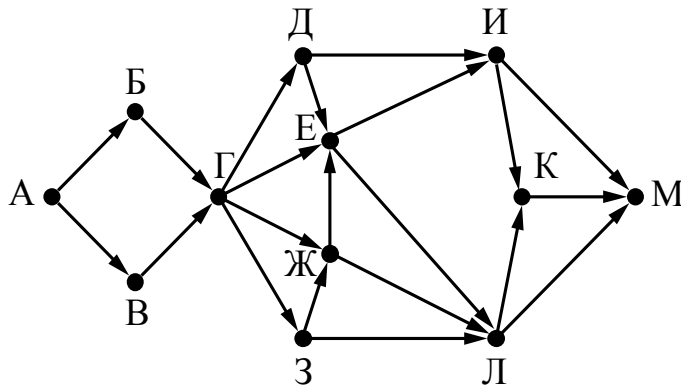
Ответ: \_\_\_\_\_.

15

На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М.

По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

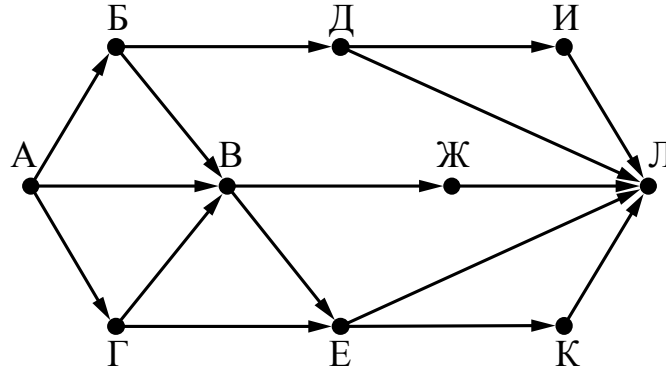
Сколько существует различных путей из города А в город М?



Ответ: \_\_\_\_\_.

16

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

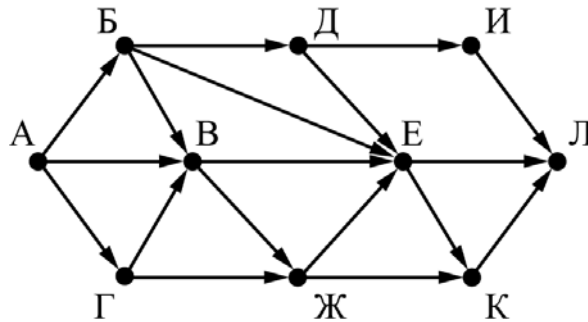


Ответ: \_\_\_\_\_.

17

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей из города А в город Л?



Ответ: \_\_\_\_\_.

## ОСНОВЫ ЛОГИКИ

18

Слава 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				0	0		0
1		1					0	1
		0						1

Каким выражением может быть  $F$ ?

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

Ответ:

19

Логическая функция  $F$  задаётся выражением

$$(\neg x \wedge y \wedge z) \vee (\neg x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z).$$

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  $x$ ,  $y$ ,  $z$ .

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

В ответе напишите буквы  $x$ ,  $y$ ,  $z$  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  $\neg x \vee y$ , зависящим от двух переменных:  $x$  и  $y$ , 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  $y$ , а второму столбцу – переменная  $x$ . В ответе следовало бы написать:  $yx$ .

Ответ: \_\_\_\_\_.

20

Логическая функция  $F$  задаётся выражением

$$(\neg x \wedge y \wedge \neg z) \vee (x \wedge y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge \neg z).$$

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  $x, y, z$ .

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

В ответе напишите буквы  $x, y, z$  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  $\neg x \vee y$ , зависящим от двух переменных:  $x$  и  $y$ , 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  $y$ , а второму столбцу – переменная  $x$ . В ответе следовало бы написать:  $yx$ .

Ответ: \_\_\_\_\_.

21

На числовой прямой даны два отрезка:  $D = [15; 40]$  и  $C = [21; 63]$ . Укажите наименьшую возможную длину такого отрезка  $A$ , что формула

$$(x \in D) \rightarrow ((\neg(x \in C) \wedge \neg(x \in A)) \rightarrow \neg(x \in D))$$

истинна (т.е. принимает значение 1 при любом значении переменной  $x$ ).

Ответ: \_\_\_\_\_.

22

На числовой прямой даны два отрезка:  $P = [30, 68]$  и  $Q = [10, 50]$ . Отрезок  $A$  таков, что формула

$$\neg(x \in A) \rightarrow ((x \in P) \rightarrow \neg(x \in Q))$$

истинна при любом значении переменной  $x$ .

Какова наименьшая возможная длина отрезка  $A$ ?

Ответ: \_\_\_\_\_.

23

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 (т.е. принимает значение 1 при любом натуральном значении переменной  $x$ )?

Ответ: \_\_\_\_\_.



24

Обозначим через  $m \& n$  поразрядную конъюнкцию неотрицательных целых чисел  $m$  и  $n$ . Так, например,  $14 \& 5 = 1110_2 \& 0101_2 = 0100_2 = 4$ .

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  $x$ )?

Ответ: \_\_\_\_\_.

25

Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_6, y_1, y_2, \dots, y_6$ , которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \vee y_1) \rightarrow (x_2 \wedge y_2) = 1$$

$$(x_2 \vee y_2) \rightarrow (x_3 \wedge y_3) = 1$$

...

$$(x_5 \vee y_5) \rightarrow (x_6 \wedge y_6) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, \dots, x_6, y_1, y_2, \dots, y_6$ , при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: \_\_\_\_\_.

26

Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$ , которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee y_1) = 1$$

$$(x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_2 \vee y_2) = 1$$

...

$$(x_6 \vee x_7) \wedge (\neg x_6 \vee \neg x_7) \wedge (\neg x_6 \vee y_6) = 1$$

$$(\neg x_7 \vee y_7) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$ , при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: \_\_\_\_\_.

**Ответы к заданиям**

№ задания	Ответ
1	2
2	10
3	7
4	7
5	6
6	6
7	5
8	29
9	2015
10	4
11	58
12	10
13	15
14	12
15	48
16	13
17	23
18	1
19	xyz
20	zxy
21	6
22	20
23	12
24	13
25	19
26	24

**РАЗДЕЛ 3**

*Ответом к заданиям 1–16 является одна цифра или число. Ответы к заданиям 1–16 запишите в отведённом месте работы.*

**Архитектура компьютеров и компьютерных сетей**

**1** 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 четырёх байтов, причём 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 111.81.176.127 адрес сети равен 111.81.160.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

**2** 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 четырёх байтов, причём 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 117.191.224.137 адрес сети равен 117.191.192.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

**3**

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 четырёх байтов, причём 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 119.134.60.57 адрес сети равен 119.134.56.0. Чему равно значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

**4**

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 четырёх байтов, причём 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 117.184.113.45 адрес сети равен 117.184.64.0. Чему равно значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

**Обработка числовой информации****5**

Дан фрагмент электронной таблицы. Из ячейки А2 в ячейку В1 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Каким стало числовое значение формулы в ячейке В1?

	<b>А</b>	<b>В</b>	<b>С</b>	<b>Д</b>	<b>Е</b>
<b>1</b>	40		400	4000	40000
<b>2</b>	=С\$2 + \$D3	3	300	3000	30000
<b>3</b>	20	2	200	2000	20000
<b>4</b>	10	1	100	1000	10000

*Примечание: знак \$ обозначает абсолютную адресацию.*

Ответ: \_\_\_\_\_.

**6**

Дан фрагмент электронной таблицы. Из ячейки Е4 в ячейку D3 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Каким стало числовое значение формулы в ячейке D3?

	<b>А</b>	<b>В</b>	<b>С</b>	<b>Д</b>	<b>Е</b>
<b>1</b>	40	5	400	70	4
<b>2</b>	30	6	300	60	3
<b>3</b>	20	7	200		2
<b>4</b>	10	8	100	40	= В3 * \$С\$2

*Примечание: знак \$ обозначает абсолютную адресацию.*

Ответ: \_\_\_\_\_.

**7**

В ячейку С1 электронной таблицы записана формула =СУММ(В1:В5) (=SUM(B1:B5)). В ячейку С2 записана формула =СРЗНАЧ(В1:В4) (=AVERAGE(B1:B4)). Какое целое число записано в ячейке В5 электронной таблицы, если известно, что значение формулы в ячейке С1 равно 30, а в ячейке С2 равно 7?

Ответ: \_\_\_\_\_.

**Базы данных**

**8**

Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных, сколько потомков мужского пола у Гуревича И.Т.

Таблица 1		
ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
	...	

Таблица 2	
ID_Родителя	ID_Ребёнка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

Ответ:

9

Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных фамилию и инициалы племянницы Лещенко П.И.

*Пояснение: племянницей считается дочь брата или сестры.*

Таблица 1		
ID	Фамилия_И.О.	Пол
14	Лещенко Н.А.	Ж
23	Лещенко И.П.	М
25	Лещенко П.И.	М
26	Лещенко П.П.	М
34	Робертс А.И.	Ж
35	Робертс В.С.	Ж
36	Робертс С.С.	М
44	Дрозд А.С.	Ж
45	Дрозд В.А.	М
46	Иванец О.С.	М
47	Иванец П.О.	М
54	Челидзе А.П.	Ж
64	Саркисян П.А.	Ж
...	...	...

Таблица 2	
ID_Родителя	ID_Ребёнка
23	25
44	25
25	26
64	26
23	34
44	34
34	35
36	35
14	36
34	46
36	46
25	54
64	54
...	...

- 1) Робертс В.С.
- 2) Иванец О.С.
- 3) Челидзе А.П.
- 4) Лещенко Н.А.

Ответ:

10

Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных, сколько сыновей и внуков мужского пола Точёновой А.А. упомянуты в таблице 1.

Таблица 1			Таблица 2	
ID	Фамилия И.О.	Пол	ID Родителя	ID Ребёнка
1141	Петрунчик Я.М.	Ж	1141	1143
1143	Петрунчик И.Б.	Ж	1141	2139
1147	Риманов Т.П.	М	1143	1147
1155	Асламоилов Ф.У.	М	1254	1141
1254	Бакст М.Б.	М	1254	2216
1270	Лом Г.Д.	Ж	1254	2133
2107	Точёнова А.А.	Ж	1270	2212
2133	Бакст Л.М.	Ж	2107	1141
2135	Ниткина А.Г.	Ж	2107	2133
2139	Фомич Н.Б.	М	2107	2216
2212	Фомич Б.В.	М	2212	2139
2216	Бакст Г.М.	М	2212	1143
2224	Монахов И.А.	М	2216	2135
...	...	...	...	...

Ответ: \_\_\_\_\_.



11

Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных суммарное число дочерей и внуков Литовченко Д.А.

Таблица 1			Таблица 2	
ID	Фамилия И.О.	Пол	ID Родителя	ID Ребёнка
1001	Пасечников А.А.	М	1001	5023
1156	Чернец Д.Д.	Ж	1001	1578
1165	Дятлова Д.Л.	Ж	1165	1798
1367	Пачка В.Д.	М	1165	1587
1578	Литовченко Д.А.	М	1578	1798
1587	Осенняя А.Д.	Ж	1578	1587
1798	Дятлов Д.Д.	М	1578	1156
2002	Литовченко К.В.	Ж	1798	9090
5023	Обручева В.А.	Ж	2002	5023
7034	Листиков П.Л.	М	2002	1578
9009	Веселых М.Л.	Ж	5023	9009
9045	Скрипка А.В.	Ж	5023	7034
9054	Веселых Л.Л.	М	9045	1001
9090	Морознова В.Л.	Ж	9054	9009
...	...	...	...	...

Ответ: \_\_\_\_\_.

**Поиск информации в сети****12**

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

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

<b>Запрос</b>	<b>Найдено страниц (в сотнях тысяч)</b>
<i>Поле</i>	45
<i>Пшеница</i>	24
<i>Напряжённость</i>	44
<i>Напряжённость   Поле   Пшеница</i>	74
<i>Напряжённость &amp; Поле</i>	19
<i>Напряжённость &amp; Пшеница</i>	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Поле & Пшеница*?

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

Ответ: \_\_\_\_\_.

**13**

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

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

<b>Запрос</b>	<b>Найдено страниц (в тысячах)</b>
<i>Евклид &amp; Аристотель</i>	240
<i>Евклид &amp; (Аристотель   Платон)</i>	450
<i>Евклид &amp; Аристотель &amp; Платон</i>	90

Компьютер печатает количество страниц (в тысячах), которое будет найдено по следующему запросу:

*Евклид & Платон*

Укажите целое число, которое напечатает компьютер.

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

Ответ: \_\_\_\_\_.

14

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

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

Запрос	Найдено страниц (в сотнях тысяч)
<i>Поле</i>	45
<i>Пшеница</i>	24
<i>Напряжённость</i>	44
<i>Поле &amp; Пшеница</i>	17
<i>Напряжённость &amp; Поле</i>	19
<i>Напряжённость &amp; Пшеница</i>	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Напряжённость | Поле | Пшеница*?

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

Ответ: \_\_\_\_\_.

15

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

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

Запрос	Найдено страниц (в тысячах)
<i>Соболь   Куница</i>	6500
<i>Куница</i>	4300
<i>Соболь</i>	3780

Какое количество страниц (в тысячах) будет найдено по запросу *Соболь & Куница*?

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

Ответ: \_\_\_\_\_.

16

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

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

Запрос	Найдено страниц (в тысячах)
<i>Пушкин &amp; Лермонтов</i>	320
<i>Пушкин &amp; Гоголь</i>	280
<i>Пушкин &amp; (Лермонтов   Гоголь)</i>	520

Какое количество страниц (в тысячах) будет найдено по запросу  
*Пушкин & Лермонтов & Гоголь*?

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

Ответ: \_\_\_\_\_.

### Ответы к заданиям

№ задания	Ответ
1	224
2	192
3	248
4	192
5	6000
6	9000
7	2
8	3
9	1
10	2
11	3
12	20
13	300
14	77
15	1580
16	80

**РАЗДЕЛ 4**

*Ответом к заданиям 1–26 является одна цифра или число. Ответы к заданиям 1–26 запишите в отведённом месте работы.*

*При выполнении заданий 27–32 используйте дополнительный лист. Ответ пишите чётко и разборчиво.*

**Алгоритмы. Элементы теории алгоритмов****1**

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .
2. К этой записи дописываются справа ещё 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.

Ответ: \_\_\_\_\_.

**2**

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .
2. К этой записи дописываются справа ещё 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 такое наименьшее число  $N$ , для которого результат работы алгоритма больше 45. В ответе это число запишите в десятичной системе счисления.

Ответ: \_\_\_\_\_.

**3**

У исполнителя Калькулятор две команды, которым присвоены номера:

1. **прибавь 2,**
2. **умножь на 5.**

Выполняя первую из них, Калькулятор прибавляет к числу на экране 2, а выполняя вторую, умножает его на 5.

*Так, для программы*

***умножь на 5,***

***прибавь 2,***

***умножь на 5,***

***прибавь 2***

*нужно написать: 2121. Эта программа преобразует число 1 в число 37.*

Запишите программу, которая преобразует **число 2 в число 24** и содержит не более четырёх команд. Указывайте лишь номера команд. Если таких программ более одной, то запишите любую из них.

Ответ: \_\_\_\_\_.

**4** У исполнителя Аккорд две команды, которым присвоены номера:

- 1. вычти 1,**
- 2. умножь на 5.**

Выполняя первую из них, Аккорд вычитает от числа на экране 1, а выполняя вторую, умножает это число на 5.

*Так, для программы*

***умножь на 5,***

***вычти 1,***

***вычти 1***

*нужно написать: 211. Эта программа преобразует число 5 в число 23.*

Запишите программу, которая преобразует **число 1 в число 99** и содержит не более пяти команд. Указывайте лишь номера команд. Если таких программ более одной, то запишите любую из них.

Ответ: \_\_\_\_\_.

**5** У исполнителя Калькулятор две команды, которым присвоены номера:

- 1. прибавь 3,**
- 2. умножь на 2.**

Выполняя первую из них, Калькулятор прибавляет к числу на экране 3, а выполняя вторую, удваивает его.

*Например, 21211 – это программа*

***умножь на 2***

***прибавь 3***

***умножь на 2***

***прибавь 3***

***прибавь 3,***

*которая преобразует число 1 в число 16.*

Запишите порядок команд в программе преобразования **числа 4 в число 37**, содержащей не более пяти команд, указывая лишь номера команд. Если таких программ более одной, то запишите любую из них.

Ответ: \_\_\_\_\_.

6

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **сместиться на  $(a, b)$** , где  $a, b$  – целые числа. Эта команда перемещает Чертёжника из точки с координатами  $(x, y)$  в точку с координатами  $(x + a, y + b)$ .

Например, если Чертёжник находится в точке с координатами  $(4, 2)$ , то команда **сместиться на  $(2, -3)$**  переместит Чертёжника 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).

Чертёжнику был дан для исполнения следующий алгоритм (число повторений и величины смещения в первой из повторяемых команд неизвестны):

НАЧАЛО

**сместиться на  $(-12, 11)$**

ПОВТОРИ ... РАЗ

**сместиться на  $(..., ...)$**

**сместиться на  $(-6, 3)$**

КОНЕЦ ПОВТОРИ

**сместиться на  $(-13, -26)$**

КОНЕЦ

В результате выполнения этого алгоритма Чертёжник возвращается в исходную точку. Какое наибольшее число повторений могло быть указано в конструкции «ПОВТОРИ ... РАЗ»?

Ответ: \_\_\_\_\_.



7

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. 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** (111, 27)

преобразует строку 05111150 в строку 0527150.

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 приведённой ниже программы к строке, состоящей из 84 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

Ответ: \_\_\_\_\_.

8

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. 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** (111, 27)

преобразует строку 05111150 в строку 0527150.

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 приведённой ниже программы к строке, состоящей из 95 идущих подряд цифр 3? В ответе запишите полученную строку.

НАЧАЛО

ПОКА **нашлось** (999) ИЛИ **нашлось** (333)

ЕСЛИ **нашлось** (999)

ТО **заменить** (999, 3)

ИНАЧЕ **заменить** (333, 9)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: \_\_\_\_\_.

9

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. 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** (111, 27)

преобразует строку 05111150 в строку 0527150.

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).

На вход приведённой ниже программы подали строку, состоящей из 28 идущих подряд цифр 9. Чему равна сумма цифр в строке, которая получится в результате выполнения программы?

НАЧАЛО

ПОКА **нашлось** (222) ИЛИ **нашлось** (999)

    ЕСЛИ **нашлось** (222)

        ТО **заменить** (222, 9)

        ИНАЧЕ **заменить** (999, 2)

    КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: \_\_\_\_\_.

10

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. 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** (111, 27)

преобразует строку 05111150 в строку 0527150.

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 приведённой ниже программы к строке, состоящей из 92 идущих подряд цифр 3? В ответе запишите полученную строку.

НАЧАЛО

ПОКА **нашлось** (999) ИЛИ **нашлось** (333)

ЕСЛИ **нашлось** (999)

ТО **заменить** (999, 3)

ИНАЧЕ **заменить** (333, 9)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: \_\_\_\_\_.

Ниже на пяти языках программирования записаны две рекурсивные функции (процедуры): F и G.

Бейсик	Python
<pre> DECLARE SUB F(n) DECLARE SUB G(n)  SUB F(n)   PRINT "*"   IF n &gt; 0 THEN G(n - 1) END SUB  SUB G(n)   PRINT "*"   IF n &gt; 0 THEN F(n - 1) END SUB                     </pre>	<pre> def F(n):     print("*")     if n &gt; 0:         G(n - 1)  def G(n):     print("*")     if n &gt; 0:         F(n - 1)                     </pre>
Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач   вывод "*"   если n &gt; 0 то     G(n - 1)   все кон  алг G(цел n) нач   вывод "*"   если n &gt; 0 то     F(n - 1)   все кон                     </pre>	<pre> procedure F(n: integer); forward; procedure G(n: integer); forward;  procedure F(n: integer); begin   writeln('*');   if n &gt; 0 then     G(n - 1); end;  procedure G(n: integer); begin   writeln('*');   if n &gt; 0 then     F(n - 1); end;                     </pre>
Си	
<pre> void F(int n); void G(int n);  void F(int n){   printf("*");   if (n &gt; 0)     G(n - 1); }  void G(int n){   printf("*");   if (n &gt; 0)     F(n - 1); }                     </pre>	

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(14)?

Ответ: \_\_\_\_\_.

**12**

Ниже на пяти языках программирования записана рекурсивная функция (процедура) F.

<b>Бейсик</b>	<b>Python</b>
<pre>SUB F(n)   print n,   IF n &gt;= 7 THEN     F(n - 3)     F(n - 1)   END IF END SUB</pre>	<pre>def F(n):   print(n, end='')   if n &gt;= 7:     F(n - 3)     F(n - 1)</pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre><u>алг</u> F(<u>цел</u> n) <u>нач</u>   <u>вывод</u> n   <u>если</u> n &gt;= 7 <u>то</u>     F(n - 3)     F(n - 1)   <u>все</u> <u>кон</u></pre>	<pre>procedure F(n: integer); begin   write(n);   if n &gt;= 7 then     begin       F(n - 3);       F(n - 1)     end end; end;</pre>
<b>Си</b>	
<pre>void F(int n) {   printf("%d", n);   if (n &gt;= 7) {     F(n - 3);     F(n - 1);   } }</pre>	

Что выведет программа при вызове F(9)? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

Ответ: \_\_\_\_\_.

**13**

Ниже на пяти языках программирования записан рекурсивный алгоритм F.

<b>Бейсик</b>	<b>Python</b>
<pre>SUB F(n)   PRINT "*"   IF n &gt; 0 THEN     F(n - 1)     F(n - 2)   END IF END SUB</pre>	<pre>def F(n):   print("*")   if n &gt; 0:     F(n - 1)     F(n - 2)</pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre><u>алг</u> F(<u>цел</u> n) <u>нач</u>   <u>вывод</u> "*"   <u>если</u> n &gt; 0 <u>то</u>     F(n - 1)     F(n - 2)   <u>все</u> <u>кон</u></pre>	<pre>procedure F(n: integer); begin   writeln('*');   if n &gt; 0 then   begin     F(n - 1);     F(n - 2)   end end</pre>
<b>Си</b>	
<pre>void F(int n) {   printf("*");   if (n &gt; 0)   {     F(n - 1);     F(n - 2);   } }</pre>	

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(4)?

Ответ: \_\_\_\_\_.

14

В программе используется одномерный целочисленный массив  $A$  с индексами от 0 до 9. Значения элементов равны 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 соответственно, т.е.  $A[0] = 9$ ,  $A[1] = 8$  и т.д.

Определите значение переменной  $s$  после выполнения следующего фрагмента этой программы (записанного ниже на пяти языках программирования).

Бейсик	Python
<pre>s = 0 FOR j = 0 TO 8   IF A(j) &gt; A(j+1) THEN     s = s + 1     t = A(j)     A(j) = A(j+1)     A(j+1) = t   END IF NEXT j</pre>	<pre>s = 0 for j in range(9):   if A[j] &gt; A[j+1]:     s = s + 1     t = A[j]     A[j] = A[j+1]     A[j+1] = t</pre>
Алгоритмический язык	Паскаль
<pre>s := 0 нц для j от 0 до 8   если A[j] &gt; A[j+1] то     s := s + 1     t := A[j]     A[j] := A[j+1]     A[j+1] := t   все кц</pre>	<pre>s := 0; for j := 0 to 8 do   if A[j] &gt; A[j+1] then   begin     s := s + 1;     t := A[j];     A[j] := A[j+1];     A[j+1] := t;   end;</pre>
Си	
<pre>s = 0; for (j = 0; j &lt; 9; j++)   if (A[j] &gt; A[j+1])   {     s++;     t = A[j];     A[j] = A[j+1];     A[j+1] = t;   }</pre>	

Ответ: \_\_\_\_\_.



**15**

В программе используется одномерный целочисленный массив *A* с индексами от 0 до 9. Значения элементов равны 4, 7, 3, 8, 5, 0, 1, 2, 9, 6 соответственно, т.е.  $A[0] = 4$ ,  $A[1] = 7$  и т.д.

Определите значение переменной *c* после выполнения следующего фрагмента этой программы (записанного ниже на пяти языках программирования).

<b>Бейсик</b>	<b>Python</b>
<pre> c = 0 FOR i = 1 TO 9   IF A(i) &lt; A(0) THEN     c = c + 1     t = A(i)     A(i) = A(0)     A(0) = t   ENDIF NEXT i                     </pre>	<pre> c = 0 for i in range(1,10):   if A[i] &lt; A[0]:     c = c + 1     t = A[i]     A[i] = A[0]     A[0] = t                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> c := 0 нц для i от 1 до 9   если A[i] &lt; A[0] то     c := c + 1     t := A[i]     A[i] := A[0]     A[0] := t   все кц                     </pre>	<pre> c := 0; for i := 1 to 9 do   if A[i] &lt; A[0] then   begin     c := c + 1;     t := A[i];     A[i] := A[0];     A[0] := t;   end;                     </pre>
<b>Си</b>	
<pre> c = 0; for (i = 1; i &lt; 10; i++)   if (A[i] &lt; A[0])   {     c++;     t = A[i];     A[i] = A[0];     A[0] = t;   }                     </pre>	

Ответ: \_\_\_\_\_.

**16**

В программе используется одномерный целочисленный массив *A* с индексами от 0 до 9. Значения элементов равны 9, 4, 2, 0, 3, 7, 1, 8, 6, 5 соответственно, т.е.  $A[0] = 9$ ,  $A[1] = 4$  и т.д.

Определите значение переменной **s** после выполнения следующего фрагмента программы (*записанного ниже на пяти языках программирования*).

<b>Бейсик</b>	<b>Python</b>
<pre>s = 0 FOR j = 0 TO 9   IF A(j) &gt; 7 THEN     s = A[j]   ENDIF NEXT j</pre>	<pre>s = 0 for j in range(10):   if A[j] &gt; 7:     s = A[j]</pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre>s := 0 нц для j от 0 до 9   если A[j] &gt; 7 то     s := A[j]   все кц</pre>	<pre>s := 0; for j := 0 to 9 do   if A[j] &gt; 7 then     s := A[j];</pre>
<b>Си</b>	
<pre>s = 0; for (j = 0; j &lt;= 9; j++)   if (A[j] &gt; 7)     s = A[j];</pre>	

Ответ: \_\_\_\_\_.

17

В программе используется одномерный целочисленный массив *A* с индексами от 0 до 9. Значения элементов равны 7, 2, 1, 5, 9, 6, 8, 4, 3, 0 соответственно, т.е.  $A[0] = 7$ ,  $A[1] = 2$  и т.д.

Определите значение переменной *s* после выполнения следующего фрагмента программы (записанного ниже на пяти языках программирования).

<p><b>Бейсик</b></p> <pre>s = 0 FOR j = 0 TO 9   IF A(j) &gt; 4 THEN     s = j   ENDIF NEXT j</pre>	<p><b>Python</b></p> <pre>s = 0 for j in range(10):   if A[j] &gt; 4:     s = j</pre>
<p><b>Алгоритмический язык</b></p> <pre>s := 0 нц для j от 0 до 9   если A[j] &gt; 4 то     s := j   все кц</pre>	<p><b>Паскаль</b></p> <pre>s := 0; for j := 0 to 9 do   if A[j] &gt; 4 then     s := j;</pre>
<p><b>Си</b></p> <pre>s = 0; for (j = 0; j &lt;= 9; j++)   if (A[j] &gt; 4)     s = j;</pre>	

Ответ: \_\_\_\_\_.

18

Ниже на пяти языках программирования записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите **наибольшее** число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 7.

Бейсик	Python
<pre> DIM X, L, M, Q AS INTEGER INPUT X Q = 8 L = 0 WHILE X &gt;= Q     L = L + 1     X = X - Q WEND M = X IF M &lt; L THEN     M = L     L = X END IF PRINT L PRINT M                     </pre>	<pre> x = int(input()) Q = 8 L = 0 while x &gt;= Q:     L = L + 1     x = x - Q M = x if M &lt; L:     M = L     L = x print(L) print(M)                     </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел x, L, M, Q     ввод x     Q := 8     L := 0     нц пока x &gt;= Q         L := L + 1         x := x - Q     кц     M := x     если M &lt; L         то             M := L             L := x         все     вывод L, M кон                     </pre>	<pre> var x, L, M, Q: integer; begin     readln(x);     Q := 8;     L := 0;     while x &gt;= Q do         begin             L := L + 1;             x := x - Q;         end;     M := x;     if M &lt; L then         begin             M := L;             L := x;         end;     writeln(L);     writeln(M); end.                     </pre>

```

Си
#include<stdio.h>
void main()
{
    int x, L, M, Q;
    scanf("%d", &x);
    Q = 8;
    L = 0;
    while (x >= Q){
        L = L + 1;
        x = x - Q;
    }
    M = x;
    if (M < L){
        M = L;
        L = x;
    }
    printf("%d\n%d", L, M);
}
    
```

Ответ: \_\_\_\_\_.

**19**

Ниже на пяти языках программирования записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите **наибольшее** число  $x$ , при вводе которого алгоритм печатает сначала 7, а потом 8.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X &gt; 0 M = M + 1 IF X MOD 2 &lt;&gt; 0 THEN L = L + 1 ENDIF X = X \ 2 WEND PRINT L PRINT M                     </pre>	<pre> x = int(input()) L = 0 M = 0 while x &gt; 0:     M = M + 1     if x % 2 != 0:         L = L + 1     x = x // 2 print(L) print(M)                     </pre>

Алгоритмический язык	Паскаль
<pre> алг нач   цел x, L, M   ввод x   L := 0   M := 0   нц пока x &gt; 0     M := M + 1     если mod(x,2) &lt;&gt; 0       то         L := L + 1       все     x := div(x,2)   кц   вывод L, нс, M кон                     </pre>	<pre> var x, L, M: integer; begin   readln(x);   L := 0;   M := 0;   while x&gt;0 do   begin     M := M+1;     if x mod 2 &lt;&gt; 0 then       L := L + 1;       x := x div 2;     end;   writeln(L);   writeln(M); end.                     </pre>
<b>Си</b>	
<pre> #include&lt;stdio.h&gt; void main() {   int x, L, M;   scanf("%d", &amp;x);   L = 0;   M = 0;   while (x &gt; 0){     M = M + 1;     if(x % 2 != 0){       L = L + 1;     }     x = x / 2;   }   printf("%d\n%d", L, M); }                     </pre>	

Ответ: \_\_\_\_\_.

20

Ниже на пяти языках программирования записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 50$ . Укажите **наименьшее** такое (т.е. большее 50) число  $x$ , при вводе которого алгоритм печатает 14.

Бейсик	Python
<pre> DIM X, L, M AS INTEGER INPUT X L = X M = 35 IF L MOD 2 = 0 THEN     M = 28 ENDIF WHILE L &lt;&gt; M IF L &gt; M THEN     L = L - M ELSE     M = M - L ENDIF WEND PRINT M                     </pre>	<pre> x = int(input()) L = x M = 35 if L % 2 == 0:     M = 28 while L != M:     if L &gt; M:         L = L - M     else:         M = M - L print(M)                     </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел x, L, M     ввод x     L := x     M := 35     если mod(L, 2) = 0         то             M := 28     все     нц пока L &lt;&gt; M         если L &gt; M             то                 L := L - M             иначе                 M := M - L         все     кц     вывод M кон                     </pre>	<pre> var x, L, M: integer; begin     readln(x);     L := x;     M := 35;     if L mod 2 = 0 then         M := 28;     while L &lt;&gt; M do         if L &gt; M then             L := L - M         else             M := M - L;         writeln(M);     end.                     </pre>

```

Си
#include<stdio.h>
void main()
{
    int x, L, M;
    scanf("%d", &x);
    L = x;
    M = 35;
    if (L % 2 == 0)
        M = 28;
    while (L != M){
        if(L > M)
            L = L - M;
        else
            M = M - L;
    }
    printf("%d", M);
}
    
```

Ответ: \_\_\_\_\_.

**21**

Ниже на пяти языках программирования записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ .

Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM X, A, B AS INTEGER INPUT X A=0: B=1 WHILE X &gt; 0     A = A+1     B = B*(X MOD 10)     X = X \ 10 WEND PRINT A PRINT B                 </pre>	<pre> x = int(input()) a = 0 b = 1 while x &gt; 0:     a = a + 1     b = b * (x % 10)     x = x // 10 print (a) print (b)                 </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел x, a, b     ввод x     a:=0; b:=1     нц пока x&gt;0         a:=a+1         b:=b*mod(x,10)         x:=div(x,10)     кц     вывод a, нс, b кон                 </pre>	<pre> var x, a, b: integer; begin     readln(x);     a:=0; b:=1;     while x&gt;0 do     begin         a:=a+1;         b:=b*(x mod 10);         x:= x div 10     end;     writeln(a); write(b) end.                 </pre>



**Си**

```
#include<stdio.h>
void main()
{
    int x, a, b;
    scanf("%d", &x);
    a=0; b=1;
    while (x>0){
        a=a+1;
        b=b*(x%10);
        x= x/10;
    }
    printf("%d\n%d", a, b);
}
```

Ответ: \_\_\_\_\_.

**22**

Исполнитель Плюс преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 2**

**2. Прибавить 4**

Первая команда увеличивает число на экране на 2, вторая увеличивает это число на 4. Программа для исполнителя Плюс – это последовательность команд.

Сколько существует программ, которые **число 9** преобразуют в **число 25**?

Ответ: \_\_\_\_\_.

**23**

Исполнитель Май36 преобразует число, записанное на экране. У исполнителя три команды, которым присвоены номера:

**1. Прибавь 1**

**2. Прибавь 3**

**3. Прибавь 6**

Первая из них увеличивает число на экране на 1, вторая увеличивает это число на 3, а третья – на 6. Программа для исполнителя Май36 – это последовательность команд.

Сколько есть программ, которые **число 141** преобразуют в **число 153**?

Ответ: \_\_\_\_\_.

**24**

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.

Ответ: \_\_\_\_\_.

**25**

Исполнитель Май15 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 Май15 – это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 21 и при этом траектория вычислений содержит число 10 и не содержит числа 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.

Ответ: \_\_\_\_\_.

**26**

Исполнитель Вычитатель преобразует число, которое записано на экране.

У исполнителя Вычитатель две команды, которым присвоены номера:

**1. Вычти 2**

**2. Вычти 5**

Первая из них уменьшает число на экране на 2, вторая уменьшает его на 5.

Программа для Вычитателя – это последовательность команд.

Сколько есть программ, которые **число 24** преобразуют в **число 3**?

Ответ: \_\_\_\_\_.

27

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от  $-10\,000$  до  $10\,000$  включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 6; 2; 9;  $-3$ ; 6 – ответ: 4.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

<b>Бейсик</b>	<b>Python</b>
<pre>CONST N AS INTEGER = 20 DIM A (1 TO N) AS INTEGER DIM I AS INTEGER,     J AS INTEGER,     K AS INTEGER  FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 20 for i in range(0, n):     a.append(int(input())) ... </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre><u>алг</u> <u>нач</u>     <u>цел</u> N = 20     <u>целтаб</u> a[1:N]     <u>цел</u> i, j, k     <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>	<pre>const     N = 20; var     a: array [1..N] of integer;     i, j, k: integer; begin     for i := 1 to N do         readln(a[i]);     ... end.</pre>

Си	Естественный язык
<pre>#include &lt;stdio.h&gt; #define N 20 int main() {     int a[N];     int i, j, k;     for (i = 0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ...     return 0; }</pre>	<p>Объявляем массив <i>A</i> из 20 элементов.                  Объявляем целочисленные переменные <i>I, J, K</i>.                  В цикле от 1 до 20 вводим элементы массива <i>A</i> с 1-го по 20-й.                  ...</p>

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 Free Pascal 2.6) 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).

28

Дан целочисленный массив из 50 элементов. Элементы массива могут принимать целые значения от 0 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число меньше 100. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 15 262 13 155 944 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Python
<pre>CONST N = 50 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER  FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre>//допускается также использовать //две целочисленные переменные j и k a = [] n = 50 for i in range(0, n):     a.append(int(input())) ...</pre>

Алгоритмический язык	Паскаль
<pre> алг нач     цел N = 50     целтаб a[1:N]     цел i, j, k     нц для i от 1 до N         ввод a[i]     кц     ... кон </pre>	<pre> const     N = 50; var     a: array [1..N] of integer;     i, j, k: integer; begin     for i := 1 to N do         readln(a[i]);     ... end. </pre>
Си	Естественный язык
<pre> #include &lt;stdio.h&gt; #define N 50 int main() {     int a[N];     int i, j, k;     for (i = 0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ...      return 0; } </pre>	<p>Объявляем массив <i>A</i> из 50 элементов.          Объявляем целочисленные переменные <i>I, J, K</i>.          В цикле от 1 до 50 вводим элементы массива <i>A</i> с 1-го по 50-й.          ...</p>

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, Free Pascal 2.6) 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).

29

Дан целочисленный массив из 50 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число двузначное. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 16 2 14 1 9 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

<b>Бейсик</b>	<b>Python</b>
<pre> CONST N = 50 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER  FOR I = 1 TO N     INPUT A(I) NEXT I ... END </pre>	<pre> //допускается также использовать //две целочисленные переменные j и k a = [] n = 50 for i in range(0, n):     a.append(int(input())) ... </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел N = 50     целтаб a[1:N]     цел i, j, k     нц для i от 1 до N         ввод a[i]     кц     ... кон </pre>	<pre> const     N = 50; var     a: array [1..N] of integer;     i, j, k: integer; begin     for i := 1 to N do         readln(a[i]);     ... end. </pre>
<b>Си</b>	<b>Естественный язык</b>
<pre> #include &lt;stdio.h&gt; #define N 50 int main() {     int a[N];     int i, j, k;     for (i = 0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ...      return 0; } </pre>	<p>Объявляем массив <i>A</i> из 50 элементов.          Объявляем целочисленные переменные <i>I, J, K</i>.          В цикле от 1 до 50 вводим элементы массива <i>A</i> с 1-го по 50-й.</p> <pre> ... </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, Free Pascal 2.6) 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).

30

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 четырёх 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, т.е. первым получивший такую позицию, что в кучах всего будет 55 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  $(4, 26)$  и  $(5, 25)$  выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

**Задание 1.** Для каждой из начальных позиций  $(4, 25)$ ,  $(6, 24)$  укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

**Задание 2.** Для каждой из начальных позиций  $(4, 24)$ ,  $(5, 24)$ ,  $(6, 23)$  укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

**Задание 3.** Для начальной позиции  $(5, 23)$  укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт 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.

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 **один** камень или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 77 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 76$ .

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 в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрывающие ходы.
- 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.

Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём 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 каждого 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 рисунке на рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.



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 **один** или **три** камня или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 18 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 36.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 36 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 35$ .

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 в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрывающие ходы.
- 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.

Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём 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 каждого 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 рисунке на рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

**Ответы к заданиям**

№ задания	Ответ
1	46
2	11
3	1211
4	21221
5	12121
6	5
7	78
8	993
9	11
10	93
11	15
12	9685746
13	15
14	9
15	2
16	8
17	6
18	61
19	254
20	70
21	98
22	34
23	17
24	28
25	14
26	10

**Критерии оценивания заданий с развёрнутым ответом**

27

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от  $-10\,000$  до  $10\,000$  включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 6; 2; 9; -3; 6 – ответ: 4.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

<b>Бейсик</b>	<b>Python</b>
<pre> CONST N AS INTEGER = 20 DIM A (1 TO N) AS INTEGER DIM I AS INTEGER,       J AS INTEGER,       K AS INTEGER  FOR I = 1 TO N     INPUT A(I) NEXT I ...  END </pre>	<pre> # допускается также # использовать две # целочисленные переменные j и k a = [] n = 20 for i in range(0, n):     a.append(int(input())) ... </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел N = 20     целтаб a[1:N]     цел i, j, k     нц для i от 1 до N         ввод a[i]     кц     ... кон </pre>	<pre> const     N = 20; var     a: array [1..N] of integer;     i, j, k: integer; begin     for i := 1 to N do         readln(a[i]);     ... end. </pre>

Си	Естественный язык
<pre>#include &lt;stdio.h&gt; #define N 20 int main() {     int a[N];     int i, j, k;     for (i = 0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ...     return 0; }</pre>	<p>Объявляем массив <i>A</i> из 20 элементов.                  Объявляем целочисленные переменные <i>I, J, K</i>.                  В цикле от 1 до 20 вводим элементы массива <i>A</i> с 1-го по 20-й.                  ...</p>

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 Free Pascal 2.6) 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).

<p align="center"><b>Содержание верного ответа и указания по оцениванию</b>                      (допускаются иные формулировки решений, приводящие к правильному результату)</p>
<p align="center"><b>На языке Паскаль</b></p>
<pre>k := 0; for i := 1 to N-1 do     if (a[i] mod 3=0) or (a[i+1] mod 3=0) then         inc(k); writeln(k);</pre>
<p align="center"><b>На алгоритмическом языке</b></p>
<pre>k := 0; нц для i от 1 до N-1     если mod(a[i],3)=0 или mod(a[i+1],3)=0         то             k := k+1     все кц вывод k</pre>
<p align="center"><b>На языке Бейсик</b></p>
<pre>K = 0 FOR I = 1 TO N-1     IF (A(I) MOD 3 = 0) OR (A(I + 1) MOD 3 = 0) THEN         K = K+1     END IF NEXT I PRINT K</pre>

<b>На языке Си</b>	
<pre>k = 0; for (i = 0; i&lt;N-1; i++)     if (a[i]%3 == 0    a[i+1]%3 == 0)         k++; printf("%d", k);</pre>	
<b>На языке Python</b>	
<pre>k = 0 for i in range(0, n - 1):     if (a[i] % 3 == 0 or a[i + 1] % 3 == 0):         k += 1 print(k)</pre>	
<b>На естественном языке</b>	
<p>Записываем в переменную <i>K</i> начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления текущего и следующего элемента массива на 3. Если первый или второй из полученных остатков равен 0, увеличиваем переменную <i>K</i> на единицу. После завершения цикла выводим значение переменной <i>K</i></p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p><i>Общие указания</i></p> <ol style="list-style-type: none"> <li>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</li> <li>2. Эффективность алгоритма не имеет значения и не оценивается.</li> <li>3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. 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</li> </ol>	
Предложен правильный алгоритм, выдающий в качестве результата верное значение	2

<p>Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее <u>не более одной</u> ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1) в цикле происходит выход за границу массива (например, используется цикл от 1 до <math>N</math>);</li> <li>2) не инициализируется или неверно инициализируется счётчик количества найденных пар;</li> <li>3) счётчик количества пар в цикле не изменяется или изменяется неверно;</li> <li>4) неверно проверяется делимость на 3;</li> <li>5) на делимость проверяются не сами элементы, а их индексы;</li> <li>6) при проверке выполнения условия для пары элементов используются неверные индексы;</li> <li>7) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении);</li> <li>8) отсутствует вывод ответа;</li> <li>9) используется переменная, не объявленная в разделе описания переменных;</li> <li>10) не указано или неверно указано условие завершения цикла;</li> <li>11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;</li> <li>12) неверно расставлены операторные скобки</li> </ol>	1
<p>Ошибок, перечисленных в п. 1–12, две или больше, или алгоритм сформулирован неверно (в том числе при отсутствии цикла в явном или неявном виде)</p>	0
<p><i>Максимальный балл</i></p>	2

28

Дан целочисленный массив из 50 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число меньше 100. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 15 262 13 155 944 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

<b>Бейсик</b>	<b>Python</b>
<pre>CONST N = 50 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER  FOR I = 1 TO N     INPUT A(I) NEXT I ...  END</pre>	<pre>//допускается также использовать //две целочисленные переменные j и k a = [] n = 50 for i in range(0, n):     a.append(int(input())) ... </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre><u>алг</u> <u>нач</u>     <u>цел</u> N = 50     <u>целтаб</u> a[1:N]     <u>цел</u> i, j, k     <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N         <u>ввод</u> a[i]     <u>кц</u>     ...  <u>кон</u></pre>	<pre>const     N = 50; var     a: array [1..N] of integer;     i, j, k: integer; begin     for i := 1 to N do         readln(a[i]);     ...  end.</pre>
<b>Си</b>	<b>Естественный язык</b>
<pre>#include &lt;stdio.h&gt; #define N 50 int main() {     int a[N];     int i, j, k;     for (i = 0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ...      return 0; }</pre>	<p>Объявляем массив <i>A</i> из 50 элементов.          Объявляем целочисленные переменные <i>I, J, K</i>.          В цикле от 1 до 50 вводим элементы массива <i>A</i> с 1-го по 50-й.          ...</p>

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, Free Pascal 2.6) 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).

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки решений, приводящие к правильному результату)
<b>На языке Паскаль</b>
<pre>k := 0; for i := 1 to N - 1 do   if (a[i]&lt;100) or (a[i+1]&lt;100) then     inc(k); writeln(k);</pre>
<b>На алгоритмическом языке</b>
<pre>k := 0; нц для i от 1 до N-1   если a[i]&lt;100 или a[i+1]&lt;100   то     k := k+1   все кц вывод k</pre>
<b>На языке Бейсик</b>
<pre>K = 0 FOR I = 1 TO N - 1   IF A(I)&lt;100 OR A(I+1)&lt;100 THEN     K = K+1   END IF NEXT I PRINT K</pre>
<b>На языке Си</b>
<pre>k = 0; for (i = 0; i &lt; N - 1; i++)   if (a[i]&lt;100    a[i+1]&lt;100)     k++; printf("%d", k);</pre>
<b>На языке Python</b>
<pre>k = 0 for i in range(0, n - 1):   if (a[i]&lt;100 or a[i+1]&lt;100):     k += 1 print(k)</pre>



<b>На естественном языке</b>	
<p>Записываем в переменную <math>K</math> начальное значение, равное 0. В цикле от первого элемента до предпоследнего сравниваем текущий и следующий элементы массива с числом 100. Если текущий или следующий элемент меньше 100, то увеличиваем переменную <math>K</math> на единицу.</p> <p>После завершения цикла выводим значение переменной <math>K</math></p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p><i>Общие указания</i></p> <p>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</p> <p>2. Эффективность алгоритма не имеет значения и не оценивается.</p> <p>3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. 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</p>	
Предложен правильный алгоритм, выдающий в качестве результата верное значение	2

<p>Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее <u>не более одной</u> ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1) в цикле происходит выход за границу массива (например, при использовании цикла от 1 до <math>N</math>);</li> <li>2) не инициализируется или неверно инициализируется счётчик количества найденных пар;</li> <li>3) счётчик количества пар в цикле не изменяется или изменяется неверно;</li> <li>4) неверно производится сравнение (например, вместо знака «меньше» используется знак «меньше или равно»);</li> <li>5) при проверке выполнения условия для пары элементов используются неверные индексы;</li> <li>6) сравнения производятся не для самих элементов массива, а для их индексов;</li> <li>7) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении);</li> <li>8) отсутствует вывод ответа;</li> <li>9) используется переменная, не объявленная в разделе описания переменных;</li> <li>10) не указано или неверно указано условие завершения цикла;</li> <li>11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;</li> <li>12) неверно расставлены операторные скобки</li> </ol>	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

29

Дан целочисленный массив из 50 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число двузначное. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 16 2 14 1 9 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

<b>Бейсик</b>	<b>Python</b>
<pre>CONST N = 50 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER  FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre>//допускается также использовать //две целочисленные переменные j и k a = [] n = 50 for i in range(0, n):     a.append(int(input())) ... </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre><u>алг</u> <u>нач</u>     <u>цел</u> N = 50     <u>целтаб</u> a[1:N]     <u>цел</u> i, j, k     <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N         <u>ввод</u> a[i]     <u>кц</u>     ... <u>кон</u></pre>	<pre>const     N = 50; var     a: array [1..N] of integer;     i, j, k: integer; begin     for i := 1 to N do         readln(a[i]);     ... end.</pre>
<b>Си</b>	<b>Естественный язык</b>
<pre>#include &lt;stdio.h&gt; #define N 50 int main() {     int a[N];     int i, j, k;     for (i = 0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ...     return 0; }</pre>	<p>Объявляем массив <i>A</i> из 50 элементов.          Объявляем целочисленные переменные <i>I, J, K</i>.          В цикле от 1 до 50 вводим элементы массива <i>A</i> с 1-го по 50-й.          ...</p>

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, Free Pascal 2.6) 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).

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки решений, приводящие к правильному результату)
<b>На языке Паскаль</b>
<pre>k := 0; for i := 1 to N - 1 do   if (a[i]&gt;=10) and (a[i]&lt;=99) or (a[i+1]&gt;=10) and (a[i+1]&lt;=99) then     inc(k); writeln(k);</pre>
<b>На алгоритмическом языке</b>
<pre>k := 0; нц для i от 1 до N-1   если a[i]&gt;=10 и a[i]&lt;=99 или a[i+1]&gt;=10 и a[i+1]&lt;=99   то     k := k+1   все кц вывод k</pre>
<b>На языке Бейсик</b>
<pre>K = 0 FOR I = 1 TO N - 1   IF A(I)&gt;=10 AND A(I)&lt;=99 OR A(I+1)&gt;=10 AND A(I+1)&lt;=99 THEN     K = K+1   END IF NEXT I PRINT K</pre>
<b>На языке Си</b>
<pre>k = 0; for (i = 0; i &lt; N - 1; i++)   if (a[i]&gt;=10 &amp;&amp; a[i]&lt;=99    a[i+1]&gt;=10 &amp;&amp; a[i+1]&lt;=99)     k++; printf("%d", k);</pre>

<b>На языке Python</b>	
<pre>k = 0 for i in range(0, n - 1):     if (a[i]&gt;=10 and a[i]&lt;=99 or a[i+1]&gt;=10 and a[i+1]&lt;=99):         k += 1 print(k)</pre>	
<b>На естественном языке</b>	
<p>Записываем в переменную <math>K</math> начальное значение, равное 0. В цикле от первого элемента до предпоследнего сравниваем текущий и следующий элементы массива с числами 10 и 99. Если хотябы один из этих элементов массива больше или равен 10 и одновременно меньше или равен 99, то увеличиваем переменную <math>K</math> на единицу.</p> <p>После завершения цикла выводим значение переменной <math>K</math></p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p><i>Общие указания</i></p> <ol style="list-style-type: none"> <li>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</li> <li>2. Эффективность алгоритма не имеет значения и не оценивается.</li> <li>3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. 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</li> </ol>	
Предложен правильный алгоритм, выдающий в качестве результата верное значение	2

<p>Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее <u>не более одной</u> ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1) в цикле происходит выход за границу массива (например, при использовании цикла от 1 до <math>N</math>);</li> <li>2) не инициализируется или неверно инициализируется счётчик количества найденных пар;</li> <li>3) счётчик количества пар в цикле не изменяется или изменяется неверно;</li> <li>4) неверно производится сравнение (например, вместо знака «больше или равно» используется знак «больше»);</li> <li>5) при проверке выполнения условия для пары элементов используются неверные индексы;</li> <li>6) сравнения производятся не для самих элементов массива, а для их индексов;</li> <li>7) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении);</li> <li>8) отсутствует вывод ответа;</li> <li>9) используется переменная, не объявленная в разделе описания переменных;</li> <li>10) не указано или неверно указано условие завершения цикла;</li> <li>11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;</li> <li>12) неверно расставлены операторные скобки</li> </ol>	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	<i>2</i>

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 четырёх 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, т.е. первым получивший такую позицию, что в кучах всего будет 55 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  $(4, 26)$  и  $(5, 25)$  выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

**Задание 1.** Для каждой из начальных позиций  $(4, 25)$ ,  $(6, 24)$  укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

**Задание 2.** Для каждой из начальных позиций  $(4, 24)$ ,  $(5, 24)$ ,  $(6, 23)$  укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

**Задание 3.** Для начальной позиции  $(5, 23)$  укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт 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.

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

**Задание 1.** В начальных позициях (4, 25), (6, 24) выигрышная стратегия есть у Вани. При начальной позиции (4, 25) после первого хода Пети может получиться одна из следующих четырёх позиций: (5, 25), (8, 25), (4, 26), (4, 50). Каждая из этих позиций содержит менее 55 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 55 камней, удвоив количество камней во второй куче. Для позиции (6, 24) после первого хода Пети может получиться одна из следующих четырёх позиций: (7, 24), (12, 24), (6, 25), (6, 48). Каждая из этих позиций содержит менее 55 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 55 камней, удвоив количество камней во второй куче. Таким образом, Ваня при любом ходе Пети выигрывает своим первым ходом.

**Задание 2.** В начальных позициях (4, 24), (5, 24) и (6, 23) выигрышная стратегия есть у Пети. При начальной позиции (4, 24) он должен первым ходом получить позицию (4, 25), из начальных позиций (5, 24) и (6, 23) Петя после первого хода должен получить позицию (6, 24). Позиции (4, 25) и (6, 24) рассмотрены при разборе задания 1. В этих позициях выигрышная стратегия есть у игрока, который будет ходить вторым (теперь это Петя). Эта стратегия описана при разборе задания 1. Таким образом, Петя при любой игре Вани выигрывает своим вторым ходом.

**Задание 3.** В начальной позиции (5, 23) выигрышная стратегия есть у Вани. После первого хода Пети может возникнуть одна из четырёх позиций: (6, 23), (5, 24), (10, 23) и (5, 46). В позициях (10, 23) и (5, 46) Ваня может выиграть одним ходом, удвоив количество камней во второй куче. Позиции (6, 23) и (5, 24) были рассмотрены при разборе задания 2. В этих позициях у игрока, который должен сделать ход (теперь это Ваня), есть выигрышная стратегия. Эта стратегия описана при разборе задания 2. Таким образом, в зависимости от игры Пети Ваня выигрывает на первом или на втором ходу.

*Примечание для эксперта.* Последняя фраза в приведённом решении избыточна. Не будет ошибкой, если экзаменуемый просто напишет, например, «При выбранной стратегии партия длится не более двух ходов».



В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом.

Исходное положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
<b>(5, 23)</b> <b>Всего: 28</b>	(5, 23+1) = (5, 24) <b>Всего: 29</b>	(5+1, 24) = (6, 24) <b>Всего: 30</b>	(6+1, 24) = (7, 24) <b>Всего: 31</b>	(7, 24*2) = (7, 48) <b>Всего: 55</b>
			(6, 24+1) = (6, 25) <b>Всего: 31</b>	(6, 25*2) = (6, 50) <b>Всего: 56</b>
			(6*2, 24) = (12, 24) <b>Всего: 36</b>	(12, 24*2) = (12, 48) <b>Всего: 60</b>
			(6, 24*2) = (6, 48) <b>Всего: 54</b>	(6, 48*2) = (6, 96) <b>Всего: 102</b>
	(5+1, 23) = (6, 23) <b>Всего: 29</b>	(6, 23+1) = (6, 24) <b>Всего: 30</b>	(6+1, 24) = (7, 24) <b>Всего: 31</b>	(7, 24*2) = (7, 48) <b>Всего: 55</b>
			(6, 24+1) = (6, 25) <b>Всего: 31</b>	(6, 25*2) = (6, 50) <b>Всего: 56</b>
			(6*2, 24) = (12, 24) <b>Всего: 36</b>	(12, 24*2) = (12, 48) <b>Всего: 60</b>
			(6, 24*2) = (6, 48) <b>Всего: 54</b>	(6, 48*2) = (6, 96) <b>Всего: 102</b>
	(5*2, 23) = (10, 23) <b>Всего: 33</b>	(10, 23*2) = (10, 46) <b>Всего: 56</b>		
	(5, 23*2) = (5, 46) <b>Всего: 51</b>	(5, 46*2) = (5, 92) <b>Всего: 97</b>		

*Примечание для эксперта.* Дерево всех партий может быть также изображено в виде ориентированного графа – так, как показано на рисунке, или другим способом. Например, вершины дерева, соответствующие одной и той же позиции, на рисунке могут быть «склеены». Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

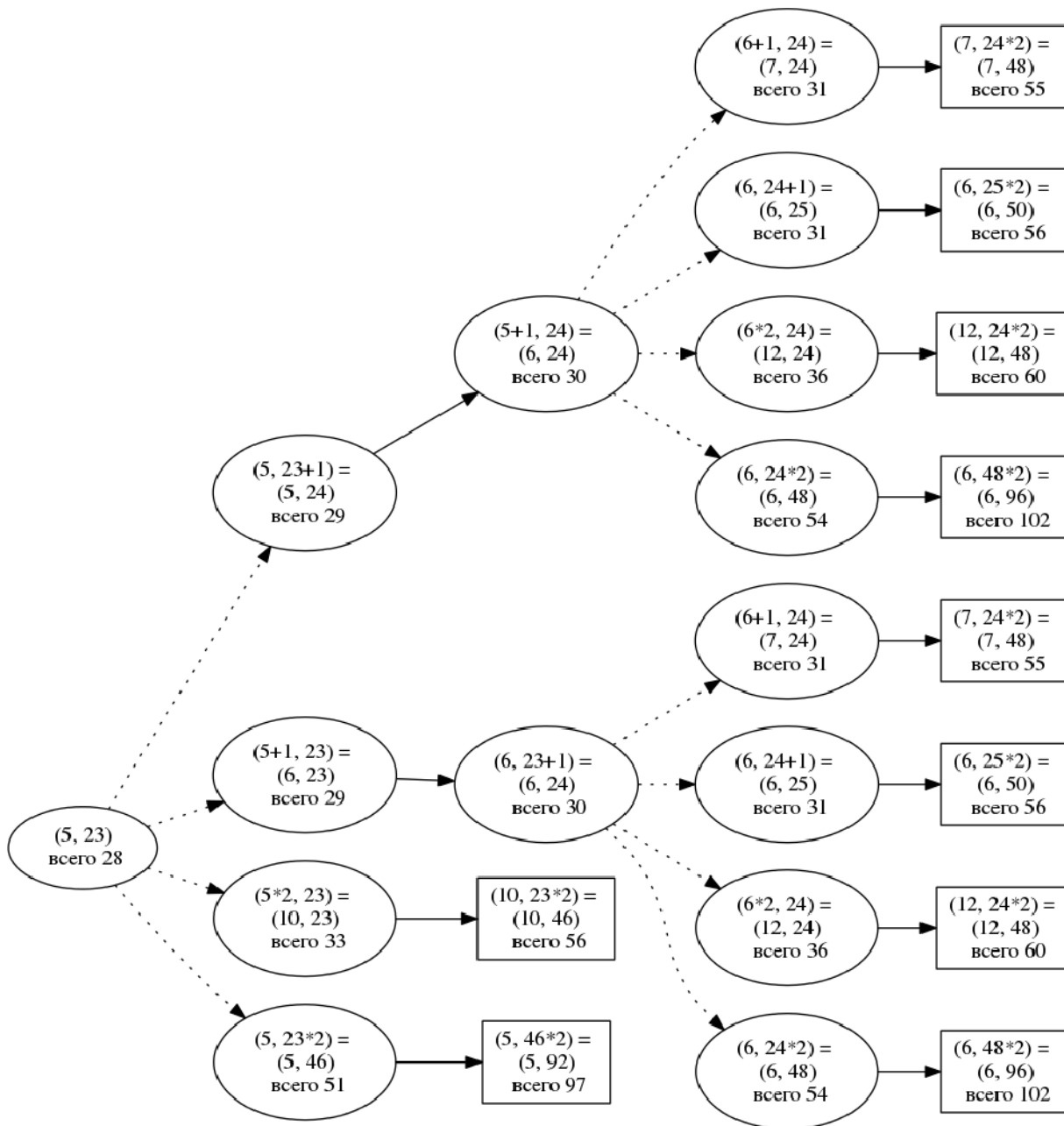


Рис. 1. Дерево всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны пунктирными стрелками, ходы Вани показаны сплошными стрелками. Заключительные позиции обозначены прямоугольником

<i>Примечание для эксперта.</i> В некоторых позициях у Вани есть и другой способ выигрыша: например, в позиции (6, 48) можно добавить один камень в любую кучу. То, что это не указано, не является ошибкой. Экзаменуемый не должен указывать все возможные выигрышные стратегии	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>В задаче от ученика требуется выполнить три задания. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом</p>	
<p>Выполнены все три задания.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий.</p> <ul style="list-style-type: none"> <li>– Выполнено задание 3.</li> <li>– Выполнены задания 1 и 2</li> </ul>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий.</p> <ul style="list-style-type: none"> <li>– Выполнено задание 1.</li> <li>– Выполнено задание 2</li> </ul>	1
<p>Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла</p>	0
<i>Максимальный балл</i>	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 **один** камень или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. Для того, чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 77 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 76$ .

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 в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрывающие ходы.
- 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.

Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём 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 каждого 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 рисунке на рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

**Задание 1**

- а) Петя может выиграть, удвоив количество камней в куче, если  $S = 39, \dots 76$ . При меньших значениях  $S$  за один ход нельзя получить кучу, в которой не менее 77 камней.
- б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет  $S = 38$  камней. Тогда после первого хода Пети в куче будет 39 камней или 76 камней. В обоих случаях Ваня удваивает количество камней и выигрывает в один ход.

*Замечание для проверяющего. В случае 76 камней игрок может выиграть и иначе – добавив один камень. В задаче не требуется указать все выигрышные стратегии. Если в работе ученика, как в приведённом примере, просто указано, что игрок всегда удваивает количество камней, это не ошибка.*

**Задание 2**

Возможные значения  $S$ : 19, 37. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 38 камней. Эта позиция разобрана в п. 1б). В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (т.е. Петя) следующим ходом выигрывает.

**Задание 3**

Возможное значение  $S$ : 36. После первого хода Пети в куче будет 37 или 72 камня. Если в куче станет 72 камня, Ваня удвоит количество камней и выигрывает первым ходом. Ситуация, когда в куче 37 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

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

Положения после очередных ходов				
и.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
<b>36</b>	$36 + 1 = 37$	$37 + 1 = 38$	$38 + 1 = 39$	<u><math>39 * 2 = 78</math></u>
			$38 * 2 = 76$	<u><math>76 + 1 = 77</math></u>
	$36 * 2 = 72$	<u><math>72 * 2 = 144</math></u>		<u><math>76 * 2 = 152</math></u>

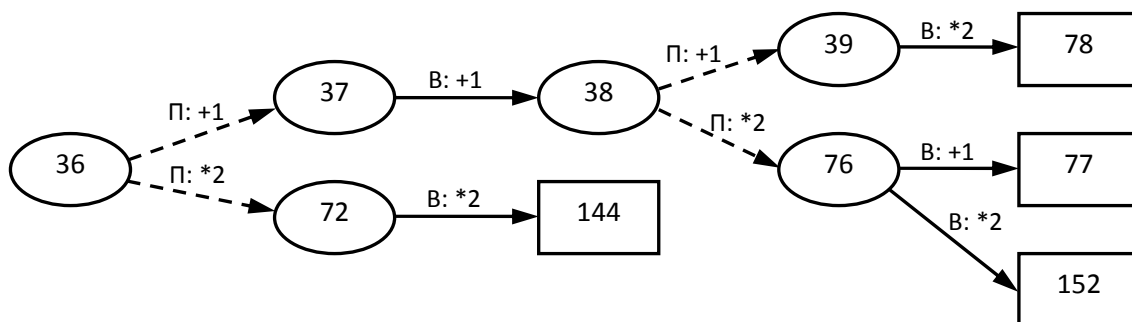


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии.

Прямоугольником обозначены позиции, в которых партия заканчивается

*Замечание для проверяющего. На рисунке для наглядности ходы Пети показаны пунктиром, а заключительные позиции выделены рамкой. И то и другое не является обязательным для экзаменуемых. Также не является ошибкой указание только одного заключительного хода Вани*

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Первое задание считается выполненным полностью, если выполнены полностью оба пункта: а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, т.е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если одновременно:</p> <ol style="list-style-type: none"> <li>1) правильно указаны все позиции, в которых Петя выигрывает первым ходом;</li> <li>2) правильно указана позиция, в которой Ваня выигрывает первым ходом;</li> <li>3) явно указано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней, но при этом не указано, каким именно ходом выигрывает Ваня.</li> </ol>	

<p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех партий, возможных при выбранной стратегии Пети.</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. В этом дереве в каждой позиции, где должен ходить Петя, должны быть разобраны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий стратегии, которую выбрал Ваня.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом</p>	
<p>Выполнены второе и третье задания.</p> <p>Первое задание выполнено полностью или частично.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ul style="list-style-type: none"> <li>• Третье задание выполнено полностью.</li> <li>• Первое и второе задания выполнены полностью.</li> <li>• Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения <math>S</math></li> </ul>	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ul style="list-style-type: none"> <li>• Первое задание выполнено полностью.</li> <li>• Во втором задании правильно указано одно из двух возможных значений <math>S</math>, и для этого значения указана и обоснована выигрышная стратегия Пети.</li> <li>• Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение <math>S</math>.</li> <li>• Для второго и третьего заданий правильно указаны значения <math>S</math></li> </ul>	1
<p>Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла</p>	0
<p><i>Максимальный балл</i></p>	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 **один** или **три** камня или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 18 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 36.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 36 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 35$ .

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 в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрывающие ходы.
- 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.

Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём 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 каждого 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 рисунке на рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.



**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

**Задание 1**

- а) Петя может выиграть, удвоив количество камней в куче, если  $S = 18, \dots, 35$ . При меньших значениях  $S$  за один ход нельзя получить кучу, в которой не менее 36 камней.
- б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет  $S = 17$  камней. Тогда после первого хода Пети в куче будет 18, 20 или 34 камня. Во всех случаях Ваня удваивает количество камней и выигрывает в один ход.

*Замечание для проверяющего. В случае 34 камней игрок может выиграть и иначе – добавив три камня. В задаче не требуется указать все выигрышные стратегии. Если в работе ученика, как в приведённом примере, просто указано, что игрок всегда удваивает количество камней, это не ошибка.*

**Задание 2**

Возможные значения  $S$ : 14, 16. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 17 камней, добавив соответственно либо три, либо один камень. Эта позиция разобрана в п. 1б). В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (т.е. Петя) следующим ходом выигрывает.

**Задание 3**

Возможные значения  $S$ : 13, 15.

Например, для  $S = 13$  после первого хода Пети в куче будет 14, 16 или 26 камней. Если в куче станет 26 камней, Ваня удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 14 или 16 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани для первого возможного значения. Для второго возможного значения дерево строится аналогично. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов				
и.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
13	$13 + 1 = 14$	$14 + 3 = 17$	$17 + 1 = 18$	<u><math>18 * 2 = 36</math></u>
			$17 + 3 = 20$	<u><math>20 * 2 = 40</math></u>
			$17 * 2 = 34$	<u><math>34 + 3 = 37</math></u>
	$13 + 3 = 16$	$16 + 1 = 17$	$17 + 1 = 18$	<u><math>18 * 2 = 36</math></u>
			$17 + 3 = 20$	<u><math>20 * 2 = 40</math></u>
			$17 * 2 = 34$	<u><math>34 + 3 = 37</math></u>
	$13 * 2 = 26$	<u><math>26 * 2 = 52</math></u>		

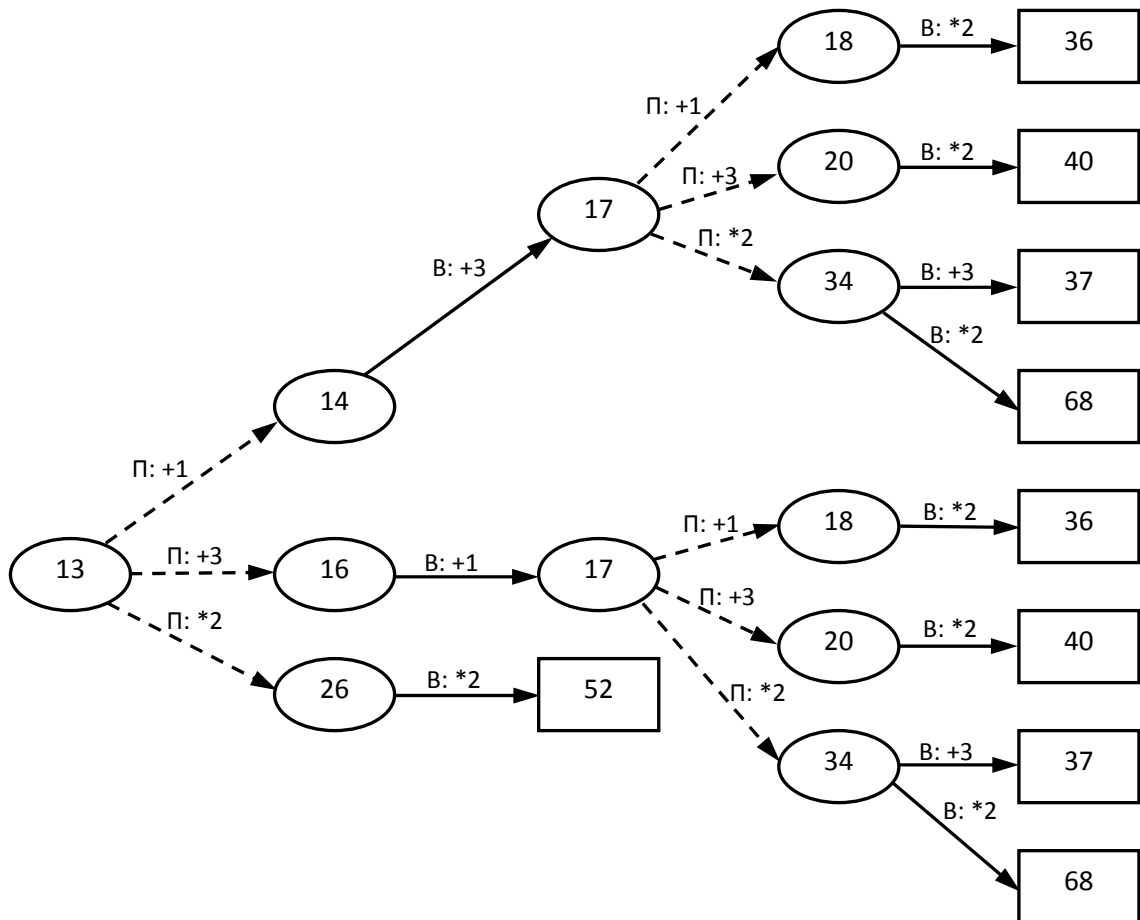


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии.

Прямоугольником обозначены позиции, в которых партия заканчивается

*Замечание для проверяющего. На рисунке для наглядности ходы Пети показаны пунктиром, а заключительные позиции выделены рамкой. И то и другое не является обязательным для экзаменуемых. Также не является ошибкой указание только одного заключительного хода Вани*

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Первое задание считается выполненным полностью, если выполнены полностью оба пункта: а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, т.е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если одновременно:</p> <ol style="list-style-type: none"> <li>1) правильно указаны все позиции, в которых Петя выигрывает первым ходом;</li> <li>2) правильно указана позиция, в которой Ваня выигрывает первым ходом;</li> <li>3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней, но при этом не указано, каким именно ходом выигрывает Ваня.</li> </ol> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех партий, возможных при выбранной стратегии Пети.</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. В этом дереве в каждой позиции, где должен ходить Петя, должны быть разобраны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий стратегии, которую выбрал Ваня.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом</p>	

<p>Выполнены второе и третье задания. Первое задание выполнено полностью или частично.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ul style="list-style-type: none"> <li>• Третье задание выполнено полностью.</li> <li>• Первое и второе задания выполнены полностью.</li> <li>• Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения <math>S</math></li> </ul>	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ul style="list-style-type: none"> <li>• Первое задание выполнено полностью.</li> <li>• Во втором задании правильно указано одно из двух возможных значений <math>S</math>, и для этого значения указана и обоснована выигрышная стратегия Пети.</li> <li>• Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение <math>S</math>.</li> <li>• Для второго и третьего заданий правильно указаны значения <math>S</math></li> </ul>	1
<p>Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла</p>	0
<i>Максимальный балл</i>	3

**РАЗДЕЛ 5**

*Ответом к заданиям 1–10 является одна цифра или число. Ответы к заданиям 1–10 запишите в отведённом месте работы. Для ответов к заданиям 11–14 используйте дополнительный лист. Ответы пишите чётко и разборчиво.*

**Программирование**

**1**

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

<p><b>Бейсик</b></p> <pre>DIM S, N AS INTEGER S = 301 N = 0 WHILE S &gt; 0     S = S - 10     N = N + 2 WEND PRINT N</pre>	<p><b>Python</b></p> <pre>s = 301 n = 0 while s &gt; 0:     s = s - 10     n = n + 2 print(n)</pre>
<p><b>Алгоритмический язык</b></p> <pre>алг нач     цел n, s     s := 301     n := 0     нц пока s &gt; 0         s := s - 10         n := n + 2     кц     вывод n кон</pre>	<p><b>Паскаль</b></p> <pre>var s, n: integer; begin     s := 301;     n := 0;     while s &gt; 0 do         begin             s := s - 10;             n := n + 2;         end;     writeln(n) end.</pre>
<p><b>Си</b></p> <pre>#include&lt;stdio.h&gt; int main() {     int s = 301, n = 0;     while (s &gt; 0) { s = s - 10; n = n + 2; }     printf("%d\n", n);     return 0; }</pre>	

Ответ: \_\_\_\_\_.

2

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM S, N AS INTEGER S = 0 N = 76 WHILE S &lt; 71     S = S + 8     N = N - 3 WEND PRINT N                     </pre>	<pre> s = 0 n = 76 while s &lt; 71:     s = s + 8     n = n - 3 print(n)                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел n, s     s := 0     n := 76     нц пока s &lt; 71         s := s + 8         n := n - 3     кц     вывод n кон                     </pre>	<pre> var s, n: integer; begin     s := 0;     n := 76;     while s &lt; 71 do         begin             s := s + 8;             n := n - 3;         end;     writeln(n) end.                     </pre>
<b>Си</b>	
<pre> #include&lt;stdio.h&gt; int main() {     int s = 0, n = 76;     while (s &lt; 71) { s = s + 8; n = n - 3; }     printf("%d\n", n);     return 0; }                     </pre>	

Ответ: \_\_\_\_\_.

3

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

<p><b>Бейсик</b></p> <pre> DIM S, N AS INTEGER S = 50 N = 1 WHILE S &gt; 0     S = S \ 4     N = N * 3 WEND PRINT N                 </pre>	<p><b>Python</b></p> <pre> s = 50 n = 1 while s &gt; 0:     s = s // 4     n = n * 3 print(n)                 </pre>
<p><b>Алгоритмический язык</b></p> <pre> алг нач     цел s, n     s := 50     n := 1     нц пока s &gt; 0         s := div(s, 4)         n := n * 3     кц     вывод n кон                 </pre>	<p><b>Паскаль</b></p> <pre> var s, n: integer; begin     s := 50;     n := 1;     while s &gt; 0 do         begin             s := s div 4;             n := n * 3;         end;     writeln(n); end.                 </pre>
<p><b>Си</b></p> <pre> #include&lt;stdio.h&gt; int main() {     int s = 50, n = 1;     while (s &gt; 0) { s = s / 4; n = n * 3; }     printf("%d\n", n);     return 0; }                 </pre>	

Ответ: \_\_\_\_\_.

4

Определите число, которое будет напечатано в результате выполнения программы, записанной ниже на пяти языках программирования.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM N, S AS INTEGER N = 10 S = 0 WHILE S &lt;= 249     S = S + 24     N = N + 3 WEND PRINT N                     </pre>	<pre> n = 10 s = 0 while s &lt;= 249:     s = s + 24     n = n + 3 print(n)                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел n, s     n := 10     s := 0     нц пока s &lt;= 249         s := s + 24         n := n + 3     кц     вывод n кон                     </pre>	<pre> var n, s: integer; begin     n := 10;     s := 0;     while s &lt;= 249 do         begin             s := s + 24;             n := n + 3;         end;     write(n) end.                     </pre>
<b>Си</b>	
<pre> #include&lt;stdio.h&gt; void main() {     int n, s;     n = 10;     s = 0;     while (s &lt;= 249)     {         s = s + 24;         n = n + 3;     }     printf("%d", n); }                     </pre>	

Ответ: \_\_\_\_\_.



**5**

Определите число, которое будет напечатано в результате выполнения программы, записанной ниже на пяти языках программирования.

<p><b>Бейсик</b></p> <pre>DIM N, S AS INTEGER N = 0 S = 0 WHILE S &lt;= 301     S = S + 10     N = N + 1 WEND PRINT N</pre>	<p><b>Python</b></p> <pre>n = 0 s = 0 while s &lt;= 301:     s = s + 10     n = n + 1 print(n)</pre>
<p><b>Алгоритмический язык</b></p> <pre>алг нач     цел n, s     n := 0     s := 0     нц пока s &lt;= 301         s := s + 10         n := n + 1     кц     вывод n кон</pre>	<p><b>Паскаль</b></p> <pre>var n, s: integer; begin     n := 0;     s := 0;     while s &lt;= 301 do         begin             s := s + 10;             n := n + 1;         end;     write(n) end.</pre>
<p><b>Си</b></p> <pre>#include&lt;stdio.h&gt; void main() {     int n, s;     n = 0;     s = 0;     while (s &lt;= 301)     {         s = s + 10;         n = n + 1;     }     printf("%d", n); }</pre>	

Ответ: \_\_\_\_\_.

6

Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма. Для Вашего удобства алгоритм представлен на пяти языках программирования.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM A, B, T, M, R AS LONG A = -20: B = 20 M = A: R = F(A) FOR T = A TO B     IF F(T) &lt;= R THEN         M = T         R = F(T)     END IF NEXT T PRINT M+R  FUNCTION F (x)     F = 2*(x*x-36)*(x*x-36)+41 END FUNCTION                     </pre>	<pre> def F(x):     return 2*(x*x-36)*(x*x-36)+41 a = -20; b = 20 M = a; R = F(a) for t in range(a,b+1):     if (F(t) &lt;= R):         M = t; R = F(t) print (M+R)                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел a, b, t, M, R     a := -20; b := 20     M := a; R := F(a)     нц для t от a до b         если F(t) &lt;= R то             M := t; R := F(t)         все     кц     вывод M+R кон алг цел F(цел x) нач     знач := 2*(x*x-36)*(x*x-36)+41 кон                     </pre>	<pre> var a, b, t, M, R :longint; function F(x: longint) : longint; begin     F := 2*(x*x-36)*(x*x-36)+41; end; begin a := -20; b := 20; M := a; R := F(a); for t := a to b do begin     if (F(t) &lt;= R) then begin         M := t;         R := F(t)     end end; write(M+R) end.                     </pre>

```

Си
#include<stdio.h>
long F(long x)
{
    return 2*(x*x-36)*(x*x-36)+41;
}
int main()
{
    long a, b, t, M, R;
    a = -20; b = 20;
    M = a; R = F(a);
    for (t=a; t<=b; t++) {
        if (F(t) <= R) {
            M = t; R = F(t);
        }
    }
    printf("%ld", M+R);
    return 0;
}
    
```

Ответ: \_\_\_\_\_.

**7**

Напишите в ответе число, равное количеству различных значений входной переменной *k*, при которых приведённая ниже программа выводит тот же ответ, что и при входном значении *k* = 10. Значение *k* = 10 также включается в подсчёт различных значений *k*. Для Вашего удобства программа приведена на пяти языках программирования.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM K, I AS LONG INPUT K I = 1 WHILE F(I) &lt; K     I = I + 1 WEND IF F(I)-K &lt;= 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) &lt; k:     i+=1 if (f(i)-k &lt;= 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) &lt; k     i := i + 1   кц   если f(i)-k &lt;= 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) &lt; k do     i := i+1;   if f(i)-k &lt;= k-f(i-1) then     writeln(i)   else     writeln(i-1); end.         </pre>
<p><b>Си</b></p>	
<pre> #include&lt;stdio.h&gt; long f(long n) {   return n * n * n; }  void main() {   long k, i;   scanf("%ld", &amp;k);   i = 1;   while (f(i)&lt;k)     i++;   if (f(i)-k &lt;= k-f(i-1)){     printf("%ld", i);   } else {     printf("%ld", i-1);   } }         </pre>	

Ответ: \_\_\_\_\_.

**8**

Напишите в ответе наименьшее значение входной переменной  $k$ , при котором программа выдаёт тот же ответ, что и при входном значении  $k = 27$ . Для Вашего удобства программа приведена на пяти языках программирования.

<b>Бейсик</b>	<b>Python</b>
<pre> DIM K, I AS LONG INPUT K I = 1 WHILE F(I) &lt; G(K)     I = I + 1 WEND PRINT I  FUNCTION F(N)     F = N * N * N END FUNCTION  FUNCTION G(N)     G = 2*N + 2 END FUNCTION                     </pre>	<pre> def f(n):     return n*n*n  def g(n):     return 2*n+2  k = int(input()) i = 1 while f(i) &lt; g(k):     i+=1 print (i)                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел i, k     ввод k     i := 1     нц пока f(i) &lt; g(k)         i := i + 1     кц     вывод i кон  алг цел f(цел n) нач     знач := n * n * n кон  алг цел g(цел n) нач     знач := 2*n + 2 кон                     </pre>	<pre> var     k, i : longint;  function f(n: longint): longint; begin     f := n * n * n; end;  function g(n: longint): longint; begin     g := 2*n + 2; end;  begin     readln(k);     i := 1;     while f(i) &lt; g(k) do         i := i+1;         writeln(i)     end.                     </pre>

**Си**

```
#include<stdio.h>
long f(long n) {
    return n * n * n;
}

long g(long n) {
    return 2*n + 2;
}

int main()
{
    long k, i;
    scanf("%ld", &k);
    i = 1;
    while(f(i)<g(k))
        i++;
    printf("%ld", i);
    return 0;
}
```

Ответ: \_\_\_\_\_.

9

Напишите в ответе число различных значений входной переменной  $k$ , при которых программа выдаёт тот же ответ, что и при входном значении  $k = 80$ . Значение  $k = 80$  также включается в подсчёт различных значений  $k$ . Для Вашего удобства программа приведена на пяти языках программирования.

<p><b>Бейсик</b></p> <pre> DIM K, I AS LONG INPUT K I = 12 WHILE I &gt; 0 AND F(I) &gt; K     I = I - 1 WEND PRINT I  FUNCTION F(N)     F = N * N END FUNCTION                 </pre>	<p><b>Python</b></p> <pre> def f(n):     return n * n  k = int(input()) i = 12 while (i &gt; 0 and f(i) &gt; k):     i = i - 1 print (i)                 </pre>
<p><b>Алгоритмический язык</b></p> <pre> алг нач     цел i, k     ввод k     i := 12     нц пока i &gt; 0 и f(i) &gt; k         i := i - 1     кц     вывод i кон алг цел f(цел n) нач     знач := n * n кон                 </pre>	<p><b>Паскаль</b></p> <pre> var k, i : longint; function f(n : longint) : longint; begin     f := n * n end;  begin     readln(k);     i := 12;     while (i&gt;0) and (f(i)&gt;k) do         i := i-1;         writeln(i)     end.                 </pre>
<p><b>Си</b></p> <pre> #include&lt;stdio.h&gt; long f(long n) {     return n * n; }  void main() {     long k, i;     scanf("%ld", &amp;k);     i = 12;     while (i&gt;0 &amp;&amp; f(i)&gt;k) do         i--;     printf("%ld", i); }                 </pre>	

Ответ: \_\_\_\_\_.

10

Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на пяти языках программирования).

<b>Бейсик</b>	<b>Python</b>
<pre> DIM A, B, T, M, R AS INTEGER A = -11: B = 11 M = A: R = F(A) FOR T = A TO B     IF F(T) &lt;= R THEN         M = T         R = F(T)     END IF NEXT T PRINT M+6  FUNCTION F(x)     F = 2*(x*x-16)*(x*x-16)+5 END FUNCTION                     </pre>	<pre> def F(x):     return 2*(x*x-16)*(x*x-16)+5  a = -11 b = 11 M = a R = F(a) for t in range(a, b+1):     if F(t) &lt;= R:         M = t         R = F(t) print (M+6)                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел a, b, t, M, R     a := -11; b := 11     M := a; R := F(a)     нц для t от a до b         если F(t) &lt;= R             то                 M := t; R := F(t)             все         кц     вывод M + 6 кон  алг цел F(цел x) нач     знач := 2*(x*x-16)*(x*x-16)+5 кон                     </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin     F := 2*(x*x-16)*(x*x-16)+5 end;  begin a := -11; b := 11; M := a; R := F(a); for t := a to b do begin     if (F(t) &lt;= R) then begin         M := t;         R := F(t)     end end; write(M+6) end.                     </pre>



```

Си
#include<stdio.h>
int F(int x)
{
    return 2*(x*x-16)*(x*x-16)+5;
}

void main()
{
    int a, b, t, M, R;
    a = -11; b = 11;
    M = a; R = F(a);
    for (t = a; t <= b; t++) {
        if (F(t) <= R) {
            M = t; R = F(t);
        }
    }
    printf("%d", M+6);
}
    
```

Ответ: \_\_\_\_\_.

**11**

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество кратных 11 чисел в исходной последовательности и минимальное кратное 11 число. Если чисел, кратных 11, нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают по абсолютной величине 1000. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

**Напоминание:** 0 делится на любое натуральное число.

<b>Бейсик</b>	<b>Python</b>
<pre> CONST n = 4 count = 0 minimum = 0 FOR I = 1 TO n     INPUT x     IF x mod 11 = 0 THEN         count = count + 1         IF x &gt; minimum THEN             minimum = x         END IF     END IF NEXT I IF count &gt; 0 THEN     PRINT count     PRINT minimum ELSE     PRINT "NO" END IF                 </pre>	<pre> n = 4 count = 0 minimum = 0 for i in range(1, n+1):     x = int(input())     if x % 11 == 0:         count += 1         if x &gt; minimum:             minimum = x if count &gt; 0:     print(count)     print(minimum) else:     print("NO")                 </pre>

Алгоритмический язык	Паскаль
<pre> алг нач   цел n = 4   цел i, x, minimum, count   count := 0   minimum := 0   нц для i от 1 до n     ввод x     если mod(x, 11) = 0 то       count := count + 1       если x &gt; minimum то         minimum := x       все     все   кц   если count &gt; 0 то     вывод count, нс     вывод minimum   иначе     вывод "NO"   все кон           </pre>	<pre> const n = 4; var i, x, minimum, count: integer; begin   count := 0;   minimum := 0;   for i := 1 to n do   begin     read(x);     if x mod 11 = 0 then     begin       count := count + 1;       if x &gt; minimum then         minimum := x       end     end;   end;   if count &gt; 0 then   begin     writeln(count);     writeln(minimum);   end   else     writeln('NO')   end.           </pre>
<b>Си</b>	
<pre> #include &lt;stdio.h&gt; #define n 4  int main() {   int i, x, minimum, count;   count = 0;   minimum = 0;   for (i = 1; i &lt;= n; i++)   {     scanf("%d", &amp;x);     if (x % 11 == 0)     {       count++;       if (x &gt; minimum)         minimum = x;     }   }   if (count &gt; 0)   {     printf("%d\n", count);     printf("%d\n", minimum);   }   else     printf("NO\n");   return 0; }           </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности: 33, 34, 44, 45.
2. Приведите пример последовательности, в которой есть хотя бы одно кратное 11 число, при вводе которой, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

**12**

Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, равное количеству цифр 3 в десятичной записи числа  $N$ . Программист торопился и допустил в программе ошибки. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования

<b>Бейсик</b>	<b>Python</b>
<pre> DIM N AS LONG DIM R, d AS INTEGER INPUT N R = 0 WHILE N &gt; 0     d = N MOD 10     IF d &lt;&gt; 3 THEN         R = R + 1     END IF     N = N \ 10 WEND PRINT d END                     </pre>	<pre> N = int(input()) R = 0 while N &gt; 0:     d = N % 10     if d != 3:         R += 1     N = N // 10 print(d)                     </pre>

Алгоритмический язык	Паскаль
<pre> алг нач   цел N, R, d   ввод N   R := 0   нц пока N &gt; 0     d := mod(N, 10)     если d &lt;&gt; 3 то       R := R + 1     все     N := div(N, 10)   кц   вывод d кон </pre>	<pre> var N: longint;     R, d: integer; begin   readln(N);   R:=0;   while N&gt;0 do begin     d := N mod 10;     if d &lt;&gt; 3 then       R := R + 1;     N := N div 10;   end;   writeln(d); end. </pre>
<b>Си</b>	
<pre> #include &lt;stdio.h&gt; int main() {   long int N;   int R, d;   scanf("%ld", &amp;N);   R = 0;   while (N &gt; 0) {     d = N % 10;     if (d != 3) {       R++;     }     N = N / 10;   }   printf("%d", d); } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 231.
2. Приведите пример входного числа  $N$ , при котором, несмотря на ошибки, приведённая программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, – приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

13

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту передаются положительные целые числа – текущие показания прибора «Сигма-2015». После того как передана серия измерений (количество измерений в серии заранее известно), прибор передаёт контрольное значение серии – наибольшее число  $R$ , удовлетворяющее следующим условиям:

- 1)  $R$  – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных, но равных по величине элементов допускаются);
- 2)  $R$  делится на 33.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную по времени и используемой памяти программу (укажите используемую версию языка программирования, например Free Pascal 2.6.4), которая будет проверять правильность контрольного значения. Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора  $N$ , 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.

Программа должна напечатать отчёт по следующей форме.

Вычисленное контрольное значение: ...

Контроль пройден (или Контроль не пройден)

Если удовлетворяющее условию контрольное значение определить невозможно, то выводится только фраза «Контроль не пройден».

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел  $N \leq 100\,000$ . В каждой из последующих  $N$  строк записано одно положительное целое число, не превышающее 1000. В последней строке записано контрольное значение.

*Пример входных данных:*

5  
66  
18  
44  
70  
29  
4620

*Пример выходных данных для приведённого выше примера входных данных:*

Вычисленное контрольное значение: 4620

Контроль пройден

14

По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности – наименьшее число  $R$ , удовлетворяющее следующим условиям:

- 1)  $R$  является произведением двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных элементов последовательности, равных по величине, допускаются);
- 2)  $R$  кратно 39.

Если такого числа  $R$  нет, то контрольное значение полагается равным 0.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет проверять правильность контрольного значения. Программа должна напечатать отчёт по следующей форме:

Вычисленное контрольное значение: ...

Контроль пройден (или – Контроль не пройден)

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел  $N$ ; в программе можно считать, что  $2 \leq N \leq 10\,000$ . В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение – натуральное число, не превышающее 1 000 000.

*Пример входных данных:*

6  
130  
39  
4  
13  
6  
300  
78

*Пример выходных данных для приведённого выше примера входных данных:*

Вычисленное контрольное значение: 78

Контроль пройден

**Ответы к заданиям**

№ задания	Ответ
1	62
2	49
3	27
4	43
5	31
6	47
7	13
8	13
9	17
10	10

**Критерии оценивания заданий с развёрнутым ответом****11**

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество кратных 11 чисел в исходной последовательности и минимальное кратное 11 число. Если чисел, кратных 11, нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают по абсолютной величине 1000. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

**Напоминание:** 0 делится на любое натуральное число.

Бейсик	Python
<pre> CONST n = 4 count = 0 minimum = 0 FOR I = 1 TO n   INPUT x   IF x mod 11 = 0 THEN     count = count + 1     IF x &gt; minimum THEN       minimum = x     END IF   END IF NEXT I IF count &gt; 0 THEN   PRINT count   PRINT minimum ELSE   PRINT "NO" END IF </pre>	<pre> n = 4 count = 0 minimum = 0 for i in range(1, n+1):     x = int(input())     if x % 11 == 0:         count += 1         if x &gt; minimum:             minimum = x if count &gt; 0:     print(count)     print(minimum) else:     print("NO") </pre>

Алгоритмический язык	Паскаль
<pre> алг нач   цел n = 4   цел i, x, minimum, count   count := 0   minimum := 0   нц для i от 1 до n     ввод x     если mod(x, 11) = 0 то       count := count + 1       если x &gt; minimum то         minimum := x       все     все   кц   если count &gt; 0 то     вывод count, нс     вывод minimum   иначе     вывод "NO"   все кон </pre>	<pre> const n = 4; var i, x, minimum, count: integer; begin   count := 0;   minimum := 0;   for i := 1 to n do     begin       read(x);       if x mod 11 = 0 then         begin           count := count + 1;           if x &gt; minimum then             minimum := x           end         end;       end;   if count &gt; 0 then     begin       writeln(count);       writeln(minimum);     end   else     writeln('NO')   end. </pre>
<b>Си</b>	
<pre> #include &lt;stdio.h&gt; #define n 4  int main() {   int i, x, minimum, count;   count = 0;   minimum = 0;   for (i = 1; i &lt;= n; i++)   {     scanf("%d", &amp;x);     if (x % 11 == 0)     {       count++;       if (x &gt; minimum)         minimum = x;     }   }   if (count &gt; 0)   {     printf("%d\n", count);     printf("%d\n", minimum);   }   else     printf("NO\n");   return 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:  
33, 34, 44, 45.
2. Приведите пример последовательности, в которой есть хотя бы одно кратное 11 число, при вводе которой, несмотря на ошибки, программа печатает правильный ответ.



3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Решение использует запись программы на Паскале. Допускается использование записи программы на любом из четырёх других языков программирования.

1. Программа выведет два числа: 2 и 44.

2. Пример последовательности, содержащей кратные 11 числа, для которой программа выдаёт правильный ответ: 51 53 55 57.

*Замечание для проверяющего. В конце работы программы значение переменной `minimum` всегда равно максимальному кратному 11 числу или 0, если в последовательности нет чисел, кратных 11. Соответственно, программа будет работать верно, если в последовательности максимальное кратное 11 число равно минимальному. Выведенное количество кратных 11 чисел будет правильным в любом случае.*

3. В программе есть две ошибки.

**Первая ошибка:** неверная инициализация ответа (переменная `minimum`).

Строка с ошибкой:

```
minimum := 0;
```

Верное исправление:

```
minimum := 1001;
```

Вместо 1001 может быть любое целое число, большее 990, либо `MAXINT`.

Можно использовать и число 990, так как при выводе мы проверяем, есть ли в последовательности хотя бы одно кратное 11 число.

**Вторая ошибка:** неверное условие перевычисления минимума.

Строка с ошибкой:

```
if x > minimum then
```

Верное исправление:

```
if x < minimum then
```

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>четыре</b> действия:</p> <ol style="list-style-type: none"> <li>1) указать, что выведет программа при конкретной входной последовательности;</li> <li>2) указать пример входной последовательности, при вводе которой программа выдаёт правильный ответ;</li> <li>3) исправить первую ошибку;</li> <li>4) исправить вторую ошибку.</li> </ol> <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:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</li> </ol>	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <ol style="list-style-type: none"> <li>а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной;</li> <li>б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки</li> </ol>	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

12

Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, равное количеству цифр 3 в десятичной записи числа  $N$ . Программист торопился и допустил в программе ошибки. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования

<b>Бейсик</b>	<b>Python</b>
<pre> DIM N AS LONG DIM R, d AS INTEGER INPUT N R = 0 WHILE N &gt; 0     d = N MOD 10     IF d &lt;&gt; 3 THEN         R = R + 1     END IF     N = N \ 10 WEND PRINT d END                     </pre>	<pre> N = int(input()) R = 0 while N &gt; 0:     d = N % 10     if d != 3:         R += 1     N = N // 10 print(d)                     </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел N, R, d     ввод N     R := 0     нц пока N &gt; 0         d := mod(N, 10)         если d &lt;&gt; 3 то             R := R + 1         все         N := div(N, 10)     кц     вывод d кон                     </pre>	<pre> var N: longint;     R, d: integer; begin     readln(N);     R:=0;     while N&gt;0 do begin         d := N mod 10;         if d &lt;&gt; 3 then             R := R + 1;         N := N div 10;     end;     writeln(d); end.                     </pre>

**Си**

```
#include <stdio.h>
int main()
{
    long int N;
    int R, d;
    scanf("%ld", &N);
    R = 0;
    while (N > 0) {
        d = N % 10;
        if (d != 3) {
            R++;
        }
        N = N / 10;
    }
    printf("%d", d);
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 231.
2. Приведите пример входного числа  $N$ , при котором, несмотря на ошибки, приведённая программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, – приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>1. Программа выведет число 2. <i>Примечание для экспертов.</i> Программа выводит значение первой слева цифры числа <math>N</math>.</p> <p>2. Программа печатает правильный ответ, например, при <math>N = 233</math>.</p> <p>3. Первая ошибка. Неверная проверка условия увеличения счётчика – переменной <math>R</math>. Строка в программе на Паскале: <code>if d&lt;&gt;3 then</code> Правильная строка: <code>if d=3 then</code></p> <p>4. Вторая ошибка. Вместо значения переменной <math>R</math> выводится значение переменной <math>d</math>. Строка в программе на Паскале: <code>writeln(d);</code> Правильная строка: <code>writeln(R);</code> <i>Примечание для экспертов.</i> Выше поясняется, в чём состоит ошибка в программе. <i>Для ученика это не обязательно.</i> <i>Достаточно указать, в какой строке была ошибка и исправленный вариант строки</i></p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>Обратите внимание! В задаче требовалось выполнить <b>четыре</b> действия:</p> <ol style="list-style-type: none"> <li>1) указать, что выведет программа при конкретном входном числе <math>N</math>;</li> <li>2) указать пример входного числа <math>N</math>, при котором программа работает правильно;</li> <li>3) исправить первую ошибку;</li> <li>4) исправить вторую ошибку.</li> </ol> <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:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</li> </ol>	

Если ученик указал только корректную исправленную строку, но при этом понятно, какую строку исходной программы он исправлял, соответствующее действие считается выполненным	
Правильно выполнены все пункты задания. Программа после исправлений для всех натуральных чисел $N$ , не превосходящих $10^9$ , выводит верный ответ. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	3
Не выполнены условия, позволяющие поставить 3 балла. При этом имеет место одна из следующих ситуаций. 1. Правильно выполнены три действия из перечисленных выше четырёх. Ни одна правильная строка не указана в качестве ошибочной. 2. Правильно выполнены все четыре действия из четырёх, одна правильная строка указана в качестве ошибочной	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место одна из следующих ситуаций. 1. Правильно выполнены два действия из перечисленных выше четырёх. Ни одна правильная строка не указана в качестве ошибочной. 2. Написана правильно работающая программа, возможно, совершенно непохожая на исходную	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

13

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту передаются положительные целые числа – текущие показания прибора «Сигма 2015». После того как передана серия измерений (количество измерений в серии заранее известно), прибор передаёт контрольное значение серии – наибольшее число  $R$ , удовлетворяющее следующим условиям:

- 1)  $R$  – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных, но равных по величине элементов допускаются);
- 2)  $R$  делится на 33.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную по времени и используемой памяти программу (укажите используемую версию языка программирования, например Free Pascal 2.6.4), которая будет проверять правильность контрольного значения. Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора  $N$ ,

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.

Программа должна напечатать отчёт по следующей форме.

Вычисленное контрольное значение: ...

Контроль пройден (или Контроль не пройден)

Если удовлетворяющее условию контрольное значение определить невозможно, то выводится только фраза «Контроль не пройден».

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел  $N \leq 100\,000$ . В каждой из последующих  $N$  строк записано одно положительное целое число, не превышающее 1000. В последней строке записано контрольное значение.

*Пример входных данных:*

5  
66  
18  
44  
70  
29  
4620

*Пример выходных данных для приведённого выше примера входных данных:*

Вычисленное контрольное значение: 4620

Контроль пройден

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

Произведение двух чисел делится на 33, если:

- один из сомножителей делится на 33 (второй может быть любым) либо
- ни один из сомножителей не делится на 33, но один из сомножителей делится на 3, а другой – на 11.

Программа, вычисляющая контрольное значение, читает все входные данные один раз, не запоминая их в массиве. Для прочитанного фрагмента входной последовательности программа хранит значения четырёх величин:

МЗ – самое большое число, кратное 3;

М11 – самое большое число, кратное 11, но не кратное 3;

М33 – самое большое число, кратное 33;

МАХ – самое большое число среди всех элементов последовательности, отличное от М33 (если число М33 встретилось более одного раза и оно является максимальным, то МАХ = М33).

После того как все данные прочитаны, искомое контрольное значение вычисляется как максимум из произведений  $М33 \cdot МАХ$  и  $МЗ \cdot М11$ .

Возможны модификации этого алгоритма. Ниже приведены программы на языках Паскаль и Си, а также на алгоритмическом языке, реализующие одну из таких модификаций. Допускаются решения, записанные на других языках программирования.



**Пример правильной и эффективной программы на языке Паскаль**

```
var M3,M11,M33,R,MAX,dat,res,i,N: longint;
begin
  M3 := 0;
  M11 := 0;
  M33 := 0;
  MAX := 0;
  readln(N);
  for i := 1 to N do
  begin
    readln(dat);
    if (dat mod 3 = 0) and (dat > M3) then
      M3 := dat
    else if (dat mod 11 = 0) and (dat > M11) then
      M11 := dat;
    if (dat mod 33 = 0) and (dat > M33) then
      begin
        if M33 > MAX then MAX := M33;
        M33 := dat
      end
    else if dat > MAX then MAX := dat;
  end;
  readln(R);
  if (M3*M11 < M33*MAX) then
    res := M33*MAX
  else
    res := M3*M11;
  if res > 0 then
    writeln('Вычисленное контрольное значение: ',res);
  if (R > 0) and (R = res)
    then writeln('Контроль пройден')
    else writeln('Контроль не пройден');
end.
```

**Пример правильной и эффективной программы на языке Си**

```
#include <stdio.h>
void main ()
{
    int N; /*количество чисел на входе*/
    int x; /*исходные данные*/
    int m3=0; /*макс. число, кратное 3*/
    int m11=0; /*макс. число, кратное 11, но не 3*/
    int m33=0; /*макс. число, кратное 33 */
    int max=0;
    int R; /*введённое контрольное значение*/
    int m; /*вычисленное контрольное значение*/
    int i;
    scanf("%d", &N);
    for (i=1; i<=N; ++i) {
        scanf("%d", &x);
        if (x % 3 == 0 && x > m3) m3 = x;
        else if (x % 11 == 0 && x > m11) m11 = x;
        if (x % 33 == 0 && x > m33) {
            if (m33 > max) max = m33;
            m33 = x;
        }
        else if (x > max) max = x;
    }
    scanf("%d", &R);
    if (m3*m11 < m33*max) m = m33*max;
    else m = m3*m11;
    if (m>0) printf("Вычисленное контрольное значение: %d\n", m);
    if (R>0 && R==m) printf("Контроль пройден\n");
    else printf("Контроль не пройден\n");
}
```

**Пример правильной и эффективной программы на алгоритмическом языке**

```
алг
нач
  цел N | количество чисел на входе
  цел x | исходные данные
  цел m3=0 | макс. число, кратное 3
  цел m11=0 | макс. число, кратное 11, но не 3
  цел m33=0 | макс. число, кратное 33
  цел mx=0
  цел R | введённое контрольное значение
  цел m | вычисленное контрольное значение
  ввод N
  нц N раз
    ввод x
    если mod(x,3) = 0 и x > m3
      то m3 := x
    иначе
      если mod(x,11) = 0 и x > m11
        то m11 := x
      все
    все
    если mod(x,33) = 0) и x > m33
      то
        если m33 > mx то mx := m33 все
        m33 := x
      иначе
        если x > mx то mx := x все
    все
  кц
  ввод R
  если m3*m11 < m33*mx
    то m := m33*mx
    иначе m := m3*m11
  все
  если m>0
    то вывод нс, "Вычисленное контрольное значение: ", m
  все
  если R>0 и R=m
    то вывод нс, "Контроль пройден"
    иначе вывод нс, "Контроль не пройден"
  все
кон
```

Указания по оцениванию	Баллы
<p>4 балла ставится за эффективную и правильно работающую программу, которая, возможно, содержит до трёх синтаксических ошибок (описок).</p> <p>3 балла ставится в случае, когда фактически задача решена, но количество описок более трёх (но не более пяти) и допущено не более одной содержательной ошибки, не позволяющей усомниться в том, что экзаменуемый правильно придумал алгоритм (список допустимых ошибок приведён ниже).</p> <p>2 балла ставится, если в дополнение к неточностям, которые перечислены выше, программа работает неэффективно по времени и/или допущено до трёх упомянутых выше содержательных ошибок. Допускается до семи описок.</p> <p>1 балл ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Далее сказанное уточнено</p>	
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> <li>1) пропущен или неверно указан знак пунктуации;</li> <li>2) неверно написано или пропущено зарезервированное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</p>	4

<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел; правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p> <p>Количество синтаксических ошибок (описок) указанных выше видов – не более пяти.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой структуре данных (контейнер <code>priority_queue</code>, <code>vector</code>, <code>set</code> или <code>map</code> в C++)).</p> <p>Допускается ошибка при вводе данных, неверный или неполный вывод результатов или неверная работа программы в «экзотических» ситуациях. Например, при использовании 16-битного целого (<code>integer</code> в WPascal или Qbasic) умножаются два числа этого типа (результат по условию может не помещаться в 16 бит).</p> <p>Кроме того, допускается наличие одной ошибки, принадлежащей к одному из следующих видов ошибок:</p> <ol style="list-style-type: none"><li>1) ошибка при инициализации максимумов;</li><li>2) неверно обрабатывается ситуация, когда один или несколько максимумов не определены;</li><li>3) неверно обрабатывается ситуация, когда максимальное произведение получается умножением одинаковых чисел (но разных элементов массива);</li><li>4) при вычислении максимумов учитываются произведения вида <code>a[i]*a[i]</code>;</li><li>5) допущен выход за границу массива;</li><li>6) используется знак "<code>&lt;</code>" вместо "<code>&lt;=</code>", "<code>or</code>" вместо "<code>and</code>" и т.п.</li></ol>	3
--	---

<p>Не выполнены условия, позволяющие поставить 3 или 4 балла.</p> <p>Программа работает в целом верно, эффективно или нет, но в реализации алгоритма есть до трёх содержательных ошибок, допустимые виды ошибок перечислены в критериях на 3 балла.</p> <p>Количество синтаксических описок не должно быть более семи.</p> <p>Программа может быть неэффективна по времени. Например, все числа запоминаются в массиве и перебираются все возможные пары элементов последовательности:</p> <pre> max := 0; for i := 1 to N - 1 do begin   for j := i + 1 to N do begin     if ((a[i]*a[j]) mod 33 = 0) and       (a[i]*a[j] &gt; max) then max := a[i]*a[j];   end end; </pre>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.</p> <p>Программа работает правильно в отдельных частных случаях.</p> <p>Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. Допускается любое количество описок</p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p><i>Максимальный балл</i></p>	4

14

По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности – наименьшее число  $R$ , удовлетворяющее следующим условиям:

- 3)  $R$  является произведением двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных элементов последовательности, равных по величине, допускаются);
- 4)  $R$  кратно 39.

Если такого числа  $R$  нет, то контрольное значение полагается равным 0.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет проверять правильность контрольного значения.

Программа должна напечатать отчёт по следующей форме:

Вычисленное контрольное значение: ...

Контроль пройден (или – Контроль не пройден)

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел  $N$ ; в программе можно считать, что  $2 \leq N \leq 10\,000$ . В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение – натуральное число, не превышающее 1 000 000.

*Пример входных данных:*

```
6
130
39
4
13
6
300
78
```

*Пример выходных данных для приведённого выше примера входных данных:*

Вычисленное контрольное значение: 78

Контроль пройден

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

Произведение двух чисел делится на 39, если выполняется одно из следующих условий:

- один из сомножителей делится на 39 (второй может быть любым);
- ни один из сомножителей не делится на 39, причём один из сомножителей делится на 13, а другой – на 3.

Поэтому программа, вычисляющая контрольное значение, может работать так.

Программа читает все входные данные один раз, не запоминая все данные в массиве. Программа для прочитанного фрагмента входной последовательности хранит значения четырёх величин:

M13 – наименьшее число, кратное 13, но не кратное 3;

M3 – наименьшее число, кратное 3, но не кратное 13;

M39 – наименьшее число, кратное 39;

MIN – наименьшее число среди всех элементов последовательности, отличное от M39 (если число M39 встретилось более одного раза и оно же является минимальным, то  $MIN = M39$ ).

После того как все данные прочитаны, искомое контрольное значение вычисляется как минимум из произведений  $M39 * MIN$  и  $M13 * M3$  (с учётом того, что значения M3, M13 или M39 могут быть не определены).

Ниже приведён пример программы на языке Паскаль, которая реализует описанный алгоритм.

Кроме того, приведён пример программы на языке Бейсик, которая правильно решает задачу, но использует алгоритм, немного отличающийся от описанного выше. Возможны и другие правильные алгоритмы.

Допускаются решения, записанные на других языках программирования



**Пример правильной и эффективной программы на языке Паскаль**

```
var M13,M3,M39,R,MIN,dat,res,i,N: longint;
begin
  M13 := 1001;
  M3 := 1001;
  M39 := 1001;
  MIN := 1001;
  readln(N);
  for i := 1 to N do
  begin
    readln(dat);
    if ((dat mod 13) = 0) and ((dat mod 3) > 0) and (dat < M13) then
      M13 := dat;
    if ((dat mod 3) = 0) and ((dat mod 13) > 0) and (dat < M3) then
      M3 := dat;
    if (dat mod 39 = 0) and (dat < M39) then
      begin
        if M39 < MIN then MIN := M39;
        M39 := dat
      end
    else
      if dat < MIN then
        MIN := dat;
    end;
    readln(R);
    if (M13 < 1001) and (M3 < 1001) then
      begin
        res := M13*M3;
        if (M39 < 1001) and (M39*MIN < res) then
          res := M39*MIN;
        end
      else
        if M39 < 1001 then
          res := M39*MIN
        else
          res:=0;

        writeln('Вычисленное контрольное значение: ',res);
        if R = res then writeln('Контроль пройден')
          else writeln('Контроль не пройден');
      end.
end.
```

**Пример правильной и эффективной программы на языке Бейсик**

```
M39 = 1001
M13 = 1001
M3 = 1001
MIN = 1001
INPUT N
FOR I = 1 TO N
  INPUT DAT
  IF DAT MOD 13 = 0 AND DAT < M13 THEN
    M13 = DAT
  ELSE
    IF DAT MOD 3 = 0 AND DAT < M3 THEN
      M3 = DAT
    END IF
  END IF
  IF DAT MOD 39 = 0 AND DAT < M39 THEN
    IF M39 < MIN THEN
      MIN = M39
    END IF
    M39 = DAT
  ELSE
    IF DAT < MIN THEN
      MIN = DAT
    END IF
  END IF
NEXT I
INPUT R
IF M13 < 1001 AND M3 < 1001 THEN
  RES = M13*M3
  IF M39 < 1001 AND M39*MIN < RES THEN
    RES = M39*MIN
  END IF
ELSE
  IF M39 < 1001 THEN
    RES = M39*MIN
  ELSE
    RES = 0
  END IF
END IF
PRINT "Вычисленное контрольное значение:"; RES
IF RES = R THEN
  PRINT "Контроль пройден"
ELSE
  PRINT "Контроль не пройден"
END IF
END
```

<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>4 балла ставится за эффективную и правильно работающую программу, которая, возможно, содержит до трёх синтаксических ошибок.</p> <p>3 балла ставится в случае, когда задача фактически решена, но программа содержит четыре-пять синтаксических ошибок, или если допущена одна содержательная ошибка, или если все входные данные сохраняются в массиве или иной структуре данных (программа неэффективна по памяти, но эффективна по времени работы).</p> <p>2 балла ставится, если программа неэффективна по времени работы (перебираются все возможные пары элементов), или в программе две содержательные ошибки, либо шесть-семь синтаксических ошибок.</p> <p>1 балл ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Далее уточняются перечисленные выше критерии</p>	
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок («описок») следующих видов (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку):</p> <ul style="list-style-type: none"> <li>– пропущен или неверно указан знак пунктуации;</li> <li>– неверно написано или пропущено зарезервированное слово языка программирования;</li> <li>– не описана или неверно описана переменная;</li> <li>– применяется операция, недопустимая для соответствующего типа данных</li> </ul>	4

<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел.</p> <p>Количество синтаксических ошибок («описок») указанных выше видов – не более пяти.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой структуре данных (контейнер <i>priority_queue</i>, <i>vector</i>, <i>set</i> или <i>map</i> в C++)).</p> <p>Допускается ошибка при вводе данных, неверный или неполный вывод результатов или неверная работа программы в «экзотических» ситуациях. Например, при использовании 16-битного целого (<i>integer</i> в <i>BPascal</i> или <i>Qbasic</i>) умножаются два числа этого типа (результат по условию может не помещаться в 16 бит).</p> <p>Кроме того, допускается наличие одной ошибки, принадлежащей к одному из следующих видов ошибок.</p> <ol style="list-style-type: none"><li>1) Допущена ошибка при инициализации минимумов.</li><li>2) Неверно обрабатывается ситуация, когда один или несколько минимумов не определены.</li><li>3) Неверно обрабатывается ситуация, когда минимальное произведение получается умножением одинаковых чисел (но разных элементов входной последовательности).</li><li>4) При вычислении минимумов учитываются произведения вида <math>a[i] * a[i]</math>.</li><li>5) Допущен выход за границу массива.</li><li>6) Допущена ошибка в реализации алгоритма сортировки.</li><li>7) Используется знак “&lt;” вместо “&lt;=”, “or” вместо “and” и т.п.</li></ol>	3
--	---

<p>Не выполнены условия, позволяющие поставить 3 или 4 балла.          Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержится до двух содержательных ошибок, допустимые виды ошибок перечислены в критериях на 3 балла.          Допускается не более семи синтаксических «описок».          Программа может быть неэффективна по времени. Например, все числа запоминаются в массиве (ниже <math>n</math> – количество прочитанных чисел), и перебираются все возможные пары элементов последовательности.          Например, так:  <pre>min := 1001*1001; for i := 1 to n - 1 do for j := i + 1 to n do   if ((a[i]*a[j]) mod 39 = 0) and     ((a[i]*a[j]) &lt; min) then     min := a[i]*a[j];</pre></p>	<p>2</p>
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.          При этом выполнено одно из двух условий.          1. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.          2. Программа правильно работает в одном из важных частных случаев, например в случае, когда искомое произведение – это произведение наименьшего числа, которое делится на 3, и наименьшего числа, которое делится на 13.          Допускается любое количество синтаксических ошибок</p>	<p>1</p>
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	<p>0</p>
<p><i>Максимальный балл</i></p>	<p>4</p>