

**Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ****Пояснения к демонстрационному варианту контрольных  
измерительных материалов единого государственного экзамена  
2014 года по ИНФОРМАТИКЕ и ИКТ**

При ознакомлении с демонстрационным вариантом контрольных измерительных материалов ЕГЭ 2014 г. следует иметь в виду, что задания, включённые в него, не отражают всех вопросов содержания, которые будут проверяться с помощью вариантов КИМ в 2014 г. Полный перечень вопросов, которые могут контролироваться на едином государственном экзамене 2014 г., приведён в кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для единого государственного экзамена 2014 г. по информатике и ИКТ.

Назначение демонстрационного варианта заключается в том, чтобы дать возможность любому участнику ЕГЭ и широкой общественности составить представление о структуре будущих КИМ, количестве заданий, об их форме, уровне сложности. Приведённые критерии оценки выполнения заданий с развёрнутым ответом, включённые в этот вариант, дают представление о требованиях к полноте и правильности записи развёрнутого ответа. Эти сведения позволят выпускникам выработать стратегию подготовки к ЕГЭ по информатике и ИКТ.

**«УТВЕРЖДАЮ»**

Руководитель

ФГБНУ «Федеральный институт  
педагогических измерений»

О. А. Решетникова

«31» октября 2013 г.

**«СОГЛАСОВАНО»**

Председатель

Научно-методического совета  
ФГБНУ «ФИПИ»  
по информатике и ИКТ

В. Б. Бетелин

«30» октября 2013 г.

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

**Демонстрационный вариант  
контрольных измерительных материалов единого  
государственного экзамена 2014 года  
по информатике и ИКТ**подготовлен Федеральным государственным бюджетным  
научным учреждением**«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»**

**Демонстрационный вариант  
контрольных измерительных материалов  
для проведения в 2014 году единого государственного экзамена  
по ИНФОРМАТИКЕ и ИКТ**

**Инструкция по выполнению работы**

На выполнение экзаменационной работы по информатике и ИКТ отводится 235 минут. Экзаменационная работа состоит из 3 частей, содержащих 32 задания. Рекомендуем не более 1,5 часа (90 минут) отвести на выполнение заданий частей 1 и 2, а остальное время – на выполнение заданий части 3.

Часть 1 содержит 13 заданий (A1–A13). К каждому заданию даётся четыре варианта ответа, из которых только один правильный.

Часть 2 состоит из 15 заданий с кратким ответом (B1–B15). К этим заданиям Вы должны самостоятельно сформулировать и записать ответ.

Часть 3 состоит из 4 заданий (C1–C4). Для выполнения заданий этой части Вам необходимо написать развёрнутый ответ в произвольной форме.

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

При выполнении заданий Вы можете пользоваться черновиком. Обращаем Ваше внимание на то, что записи в черновике не будут учитываться при оценивании работы.

Советуем выполнять задания в том порядке, в котором они даны. Для экономии времени пропускайте задание, которое не удаётся выполнить сразу, и переходите к следующему. Если после выполнения всей работы у Вас останется время, Вы сможете вернуться к пропущенным заданиям.

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

**Желаем успеха!**

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

- a) *отрицание* (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
- b) *конъюнкция* (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A \& B$ );
- c) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ ) либо  $|$  (например,  $A | B$ );
- d) *следование* (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
- e) *тождество* обозначается  $\equiv$  (например,  $A \equiv B$ ). Выражение  $A \equiv B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают (либо они оба истинны, либо они оба ложны);
- f) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

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

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

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

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

**Часть 1**

*При выполнении заданий этой части в бланке ответов № 1 под номером выполняемого Вами задания (A1–A13) поставьте знак «x» в клеточке, номер которой соответствует номеру выбранного Вами ответа.*

**A1** Дано  $N = 227_8$ ,  $M = 99_{16}$ . Какое из чисел  $K$ , записанных в двоичной системе, отвечает условию  $N < K < M$ ?

- 1)  $10011001_2$     2)  $10011100_2$     3)  $10000110_2$     4)  $10011000_2$

**A2** Между населёнными пунктами A, B, C, 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	4	6		16
B	2			3		
C	4			3		
D	6	3	3		4	9
E				4		3
F	16			9	3	

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

- 1) 12    2) 13    3) 14    4) 16

**A3** Дан фрагмент таблицы истинности выражения F.

x1	x2	x3	x4	x5	x6	x7	x8	F
1	1	0	1	1	1	1	1	0
1	0	1	0	1	1	0	1	1
0	1	0	1	1	0	1	1	1

Каким выражением может быть F?

- 1)  $\neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge x5 \wedge \neg x6 \wedge x7 \wedge x8$   
 2)  $\neg x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee \neg x6 \vee \neg x7 \vee \neg x8$   
 3)  $x1 \wedge \neg x2 \wedge x3 \wedge \neg x4 \wedge x5 \wedge x6 \wedge \neg x7 \wedge x8$   
 4)  $x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7 \vee \neg x8$

**A4** Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы.

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «\*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «\*» может задавать и пустую последовательность.

Определите, по какой из масок может быть выбрана указанная группа файлов:

comfort.c  
cobalt.cpp  
coat.c  
cost.cpp

- 1)  $co?t.c?$     2)  $c*.c*$     3)  $co*t.c??$     4)  $c*.???$

**A5** В некоторой информационной системе информация кодируется двоичными шестиразрядными словами. При передаче данных возможны их искажения, поэтому в конце 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 принятое слово не изменяется.

Исходное сообщение  
1100101 1001011 0011000  
было принято в виде  
1100111 1001110 0011000.

Как будет выглядеть принятое сообщение после обработки?

- 1) 1100111 1001011 0011000  
 2) 1100111 1001110 0000000  
 3) 0000000 0000000 0011000  
 4) 0000000 1001110 0011000

**A6** 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 дяди Гресс О.С.

*Пояснение: дядей считается родной брат отца или матери.*

Таблица 1		
ID	Фамилия И.О.	Пол
14	Грач Н.А.	Ж
24	Петренко И.П.	М
25	Петренко П.И.	М
26	Петренко П.П.	М
34	Ерёма А.И.	Ж
35	Ерёма В.С.	Ж
36	Ерёма С.С.	М
44	Лебедь А.С.	Ж
45	Лебедь В.А.	М
46	Гресс О.С.	Ж
47	Гресс П.О.	М
54	Клычко А.П.	Ж
64	Крот П.А.	Ж
...	...	...

Таблица 2	
ID Родителя	ID Ребёнка
24	25
44	25
25	26
64	26
24	34
44	34
34	35
36	35
14	36
34	46
36	46
25	54
64	54
...	...

- 1) Петренко И.П. 2) Петренко П.И. 3) Лебедь В.А. 4) Гресс П.О.

**A7** Коле нужно с помощью электронных таблиц построить таблицу двузначных чисел от 10 до 49.

Для этого сначала в диапазоне В1:К1 он записал числа от 0 до 9, и в диапазоне А2:А5 он записал числа от 1 до 4. Затем в ячейку В2 записал формулу двузначного числа (А2 – число десятков; В1 – число единиц), после чего скопировал её во все ячейки диапазона В2:К5. В итоге получил таблицу двузначных чисел. На рисунке ниже представлен фрагмент этой таблицы.

	А	В	С	Д	Е
1		0	1	2	3
2	1	10	11	12	13
3	2	20	21	22	23
4	3	30	31	32	33
5	4	40	41	42	43

Какая формула была записана в ячейке В2?

- 1) =A2\*10+\$B1 2) =A\$2\*10+\$B1 3) =\$A2\*10+B\$1 4) =A2\*10+B1

**A8** Производится четырёхканальная (квадро) звукозапись с частотой дискретизации 48 кГц и 32-битным разрешением. Запись длится 2 минуты, её результаты записываются в файл, сжатие данных не производится. Какая из приведённых ниже величин наиболее близка к размеру полученного файла?

- 1) 15 Мбайт 2) 27 Мбайт 3) 42 Мбайт 4) 88 Мбайт

**A9** Для передачи данных по каналу связи используется 5-битовый код. Сообщение содержит только буквы А, Б и В, которые кодируются следующими кодовыми словами:

А – 11010, Б – 00110, В – 10101.

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 10110, 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 (она обозначается 'х').

Получено сообщение 00111 11110 11000 10111. Декодируйте это сообщение – выберите правильный вариант.

- 1) БААВ 2) БААх 3) хххх 4) хААх

**A10** На числовой прямой даны два отрезка: P = [1, 39] и Q = [23, 58]. Выберите из предложенных отрезков такой отрезок А, что логическое выражение

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

тождественно истинно, то есть принимает значение 1 при любом значении переменной х.

- 1) [5, 20] 2) [25, 35] 3) [40, 55] 4) [20, 40]

**A11** В велокроссе участвуют 359 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Какой объём памяти будет использован устройством, когда промежуточный финиш прошли 168 велосипедистов?

- 1) 168 бит 2) 168 байт 3) 189 байт 4) 359 байт

**A12** Ниже приведён фрагмент программы, записанный на четырёх языках программирования. Массив *A* одномерный; в программе рассматривается его фрагмент, соответствующий значениям индекса от 1 до *n*.

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

Чему будет равно значение переменной *s* после выполнения данного фрагмента программы?

- 1) минимальному элементу в массиве *A*
- 2) индексу минимального элемента в массиве *A* (наименьшему из таких индексов, если минимальных элементов несколько)
- 3) индексу минимального элемента в массиве *A* (наибольшему из таких индексов, если минимальных элементов несколько)
- 4) количеству элементов, равных минимальному в массиве *A*

**A13** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, включает в себя 4 команды-приказа и 4 команды – проверки условия. Команды-приказы:

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

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится, и программа прервётся. Другие 4 команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

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

Цикл

ПОКА *условие*  
*последовательность команд*

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*  
 ТО *команда1*  
 ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка A1)?

НАЧАЛО

ПОКА **слева свободно** ИЛИ **сверху свободно**

ЕСЛИ **слева свободно**

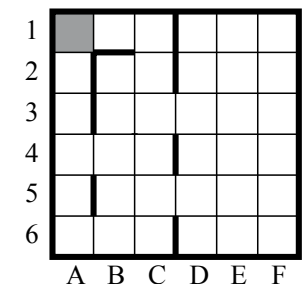
ТО **влево**

ИНАЧЕ **вверх**

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ



- 1) 14
- 2) 18
- 3) 20
- 4) 22

**Часть 2**

*Ответом к заданиям этой части (В1–В15) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишете в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**В1** У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат,
2. прибавь 1.

Первая из них возводит число на экране в квадрат, вторая увеличивает его на 1. Запишите порядок команд в программе, которая преобразует число 1 в число 10 и содержит не более 4 команд. Указывайте лишь номера команд.

(Например, программа **2122** – это программа

- прибавь 1,**  
**возведи в квадрат,**  
**прибавь 1,**  
**прибавь 1.**

Эта программа преобразует число 3 в число 18.)

Ответ: \_\_\_\_\_.

**В2** Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

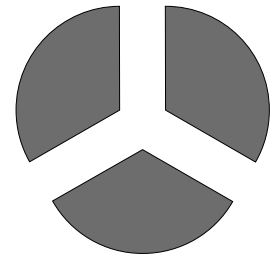
Бейсик	Паскаль
<pre>a = 25 b = 12 a = 3 * b - a IF a &gt; b THEN   c = 2 * a - b ELSE   c = 2 * a + b ENDIF</pre>	<pre>a := 25; b := 12; a := 3 * b - a; if a &gt; b then   c := 2 * a - b else   c := 2 * a + b;</pre>
Си	Алгоритмический
<pre>a = 25; b = 12; a = 3 * b - a; if (a &gt; b)   c = 2 * a - b; else   c = 2 * a + b;</pre>	<pre>a := 25 b := 12 a := 3 * b - a если a &gt; b   то c := 2 * a - b   иначе c := 2 * a + b все</pre>

Ответ: \_\_\_\_\_.

**В3** Дан фрагмент электронной таблицы.

	A	B	C
1	8		6
2	$= (B1+1)/(2*A1)$	$= 1/(B1+1)$	$= 3/(2*B1+C1)$

Какое целое число должно быть записано в ячейке В1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек А2:С2 соответствовала рисунку?



Известно, что все значения диапазона, по которым построена диаграмма, имеют один и тот же знак.

Ответ: \_\_\_\_\_.

**В4** Для передачи аварийных сигналов договорились использовать специальные цветные сигнальные ракеты, запускаемые последовательно. Одна последовательность ракет – один сигнал; в каком порядке идут цвета – существенно. Какое количество различных сигналов можно передать при помощи запуска ровно пяти таких сигнальных ракет, если в запасе имеются ракеты трёх различных цветов (ракет каждого вида неограниченное количество, цвет ракет в последовательности может повторяться)?

Ответ: \_\_\_\_\_.

**B5** Запишите число, которое будет напечатано в результате выполнения программы (для Вашего удобства программа представлена на четырёх языках программирования).

Бейсик	Паскаль
<pre>DIM N, S AS INTEGER N = 0 S = 0 WHILE S &lt;= 325   S = S + 10   N = N + 3 WEND PRINT N</pre>	<pre>var n, s: integer; begin   n := 0;   s := 0;   while s &lt;= 325 do     begin       s := s + 10;       n := n + 3     end;   write(n) end.</pre>
Си	Алгоритмический
<pre>#include&lt;stdio.h&gt; void main() {   int n, s;   n = 0;   s = 0;   while (s &lt;= 325)   {     s = s + 10;     n = n + 3;   }   printf("%d", n); }</pre>	<pre>алг нач   цел n, s   n := 0   s := 0   <u>нц пока</u> s &lt;= 325     s := s + 10     n := n + 3   <u>кц</u>   <u>вывод</u> n <u>кон</u></pre>

Ответ: \_\_\_\_\_.

**B6** Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n \leq 2;$$

$$F(n) = F(n - 1) + 2 \times F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции  $F(7)$ ?

В ответе запишите только натуральное число.

Ответ: \_\_\_\_\_.

**B7** Укажите наименьшее основание системы счисления, в которой запись десятичного числа 30 имеет ровно три значащих разряда.

Ответ: \_\_\_\_\_.

**B8** Ниже на четырёх языках записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 13, а потом 5.

Бейсик	Паскаль
<pre>DIM X, A, B, C AS INTEGER INPUT X A = 0: B = 10 WHILE X &gt; 0   C = X MOD 10   A = A + C   IF C &lt; B THEN B = C   X = X \ 10 WEND PRINT A PRINT B</pre>	<pre>var x, a, b, c: integer; begin   readln(x);   a := 0; b := 10;   while x&gt;0 do     begin       c := x mod 10;       a := a+c;       if c&lt;b then b := c;       x := x div 10;     end;   writeln(a); write(b); end.</pre>
Си	Алгоритмический
<pre>#include&lt;stdio.h&gt; void main() {   int x, a, b, c;   scanf("%d", &amp;x);   a = 0; b = 10;   while (x&gt;0) {     c = x%10;     a = a+c;     if (c&lt;b)       b = c;     x = x/10;   }   printf("%d\n%d", a, b); }</pre>	<pre><u>алг</u> <u>нач</u>   <u>цел</u> x, a, b, c   <u>ввод</u> x   a := 0; b := 10   <u>нц пока</u> x&gt;0     c := mod(x,10)     a := a+c     <u>если</u> c&lt;b       <u>то</u> b := c     <u>все</u>     x := div(x,10)   <u>кц</u>   <u>вывод</u> a, <u>нс</u>, b <u>кон</u></pre>

Ответ: \_\_\_\_\_.





**B13** У исполнителя Удвоитель две команды, которым присвоены номера:

**1. прибавь 1,**

**2. умножь на 2.**

Первая из них увеличивает число на экране на 1, вторая удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 22?

Ответ: \_\_\_\_\_.

**B14** Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх языках).

Бейсик	Паскаль
<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> 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	Алгоритмический
<pre> #include&lt;stdio.h&gt; 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 &lt;= b; t++) {         if (F(t) &lt; R) {             M = t; R = F(t);         }     }     printf("%d", M+6); }                     </pre>	<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>

Ответ: \_\_\_\_\_.

**B15** Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_{10}$ , которые удовлетворяют всем перечисленным ниже условиям?

$$\neg(x_1 \equiv x_2) \wedge ((x_1 \wedge \neg x_3) \vee (\neg x_1 \wedge x_3)) = 0$$

$$\neg(x_2 \equiv x_3) \wedge ((x_2 \wedge \neg x_4) \vee (\neg x_2 \wedge x_4)) = 0$$

...

$$\neg(x_8 \equiv x_9) \wedge ((x_8 \wedge \neg x_{10}) \vee (\neg x_8 \wedge x_{10})) = 0$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, \dots, x_{10}$  при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: \_\_\_\_\_.

*Не забудьте перенести все ответы в бланк ответов № 1.*

**Часть 3**

*Для записи ответов на задания этой части (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1, C2 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**C1**

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится максимальная цифра этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках программирования.)

<b>Бейсик</b>	<b>Паскаль</b>
<pre>DIM N AS LONG INPUT N max_digit = 9 WHILE N &gt;= 10   digit = N MOD 10   IF digit &gt; max_digit THEN     max_digit = digit   END IF   N = N \ 10 WEND PRINT max_digit END</pre>	<pre>var N: longint;     digit, max_digit: integer; begin   readln(N);   max_digit := 9;   while N &gt;= 10 do     begin       digit := N mod 10;       if digit &gt; max_digit then         max_digit := digit;       N := N div 10;     end;   writeln(max_digit); end.</pre>
<b>Си</b>	<b>Алгоритмический язык</b>
<pre>#include&lt;stdio.h&gt; int main() {   long int N;   int digit, max_digit;   scanf("%ld", &amp;N);   max_digit = 9;   while (N &gt;= 10)   {     digit = N % 10;     if (digit &gt; max_digit)       max_digit = digit;     N = N /10;   }   printf("%d", max_digit); }</pre>	<pre><u>алг</u> <u>нач</u>   <u>цел</u> N, digit, max_digit   <u>ввод</u> N   max_digit := 9   <u>нц пока</u> N &gt;= 10     digit := mod(N, 10)     <u>если</u> digit &gt; max_digit <u>то</u>       max_digit := digit   <u>все</u>   N := div(N, 10) <u>кц</u>   <u>вывод</u> max_digit <u>кон</u></pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 423.
2. 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. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

**C2**

Дан целочисленный массив из 20 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 максимальное значение среди трёхзначных элементов массива, не делящихся на 9. Если в исходном массиве нет элемента, значение которого является трёхзначным числом и при этом не кратно 9, то выведите сообщение «Не найдено».

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	Паскаль
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, MAX AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I  ...  END</pre>	<pre>const     N = 20; var     a: array [1..N] of integer;     i, j, max: integer; begin     for i := 1 to N do         readln(a[i]);     ... end.</pre>
Си	Алгоритмический язык
<pre>#include &lt;stdio.h&gt; #define N 20 void main() {     int a[N];     int i, j, max;     for (i = 0; i &lt; N; i++)         scanf("%d", &amp;a[i]);     ... }</pre>	<pre>алг нач     цел N = 20     целтаб a[1:N]     цел i, j, max     <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N         <u>ввод</u> a[i]     <u>кц</u>     ... кон</pre>
Естественный язык	
<p>Объявляем массив A из 20 элементов.                  Объявляем целочисленные переменные I, J, MAX.                  В цикле от 1 до 20 вводим элементы массива A 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 2.4) 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).

**C3**

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 **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 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 27. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 27 или больше камней.

В начальный момент в куче было  $S$  камней,  $1 \leq S \leq 26$ .

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.

- а) Укажите все такие значения числа  $S$ , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрышающий ход для каждого указанного значения  $S$ .  
 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.
- Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.
- Укажите значение  $S$ , при котором:
  - у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и
  - 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 рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в куче.

**C4**

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 – наибольшее число  $R$ , удовлетворяющее следующим условиям:

- 1)  $R$  – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел; допускаются произведения различных элементов последовательности, равных по величине);
- 2)  $R$  делится на 21.

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 каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение.

*Пример входных данных:*

6  
70  
21  
997  
7  
9  
300  
21000

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

Вычисленное контрольное значение: 21000  
Контроль пройден

**Система оценивания экзаменационной работы по информатике и ИКТ**

**Часть 1**

За правильный ответ на каждое задание части 1 ставится 1 балл.

Если указаны два и более ответов (в том числе правильный), неверный ответ или ответ отсутствует – 0 баллов.

№ задания	Ответ
A1	4
A2	1
A3	2
A4	2
A5	4
A6	2
A7	3
A8	4
A9	1
A10	2
A11	3
A12	2
A13	3

**Часть 2**

За правильный ответ на каждое задание части 2 ставится 1 балл; за неверный ответ или его отсутствие – 0 баллов.

№ задания	Ответ
B1	2212
B2	34
B3	3
B4	243
B5	99
B6	43
B7	4
B8	58
B9	23
B10	A33
B11	BCDA
B12	410
B13	37
B14	2
B15	20

**Часть 3**

**Критерии оценивания заданий с развёрнутым ответом**

**C1**

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N, не превосходящее  $10^9$ , и выводится максимальная цифра этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках программирования.)

<b>Бейсик</b>	<b>Паскаль</b>
<pre>DIM N AS LONG INPUT N max_digit = 9 WHILE N &gt;= 10   digit = N MOD 10   IF digit &gt; max_digit THEN     max_digit = digit   END IF   N = N \ 10 WEND PRINT max_digit END</pre>	<pre>var N: longint;     digit, max_digit: integer; begin   readln(N);   max_digit := 9;   while N &gt;= 10 do     begin       digit := N mod 10;       if digit &gt; max_digit then         max_digit := digit;       N := N div 10;     end;   writeln(max_digit); end.</pre>
<b>Си</b>	<b>Алгоритмический язык</b>
<pre>#include&lt;stdio.h&gt; int main() {   long int N;   int digit, max_digit;   scanf("%ld", &amp;N);   max_digit = 9;   while (N &gt;= 10)   {     digit = N % 10;     if (digit &gt; max_digit)       max_digit = digit;     N = N /10;   }   printf("%d", max_digit); }</pre>	<pre>алг нач   цел N, digit, max_digit   ввод N   max_digit := 9   нц пока N &gt;= 10     digit := mod(N, 10)     если digit &gt; max_digit то       max_digit := digit     все     N := div(N, 10)   кц   вывод max_digit кон</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 423.
2. 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. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.</p> <p>1. Программа выведет число 9.</p> <p>2. Первая ошибка. Неверная инициализация ответа (переменная max_digit). Строка с ошибкой: max_digit := 9; Возможные варианты исправления: max_digit := 0; Возможны и другие исправления инициализации, например на отрицательное число, в том числе -maxint.</p> <p>3. Вторая ошибка. Неверное условие продолжения цикла. Программа не будет рассматривать старшую цифру числа. Строка с ошибкой: while N &gt;= 10 do Возможные варианты исправления: while (N &gt;= 1) do или while (N &gt; 0) do При этом замены на while (N &gt; 1) do или while (N &gt;= 0) do корректными не являются</p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>Обратите внимание! В задаче требовалось выполнить <b>три</b> действия: указать, что выведет программа при конкретном входном значении, и исправить две ошибки.</p> <p>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия (ниже указано, какое действие считается выполненным).</p> <p>1. Верно указано, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указана и верно исправлена ошибка инициализации (не обязательно с упоминанием этого термина).</p> <p>3. Указано на неверное условие продолжения цикла, и оно исправлено на верное.</p> <p>Каждый из п. 2 и 3 считается выполненным, если:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> </ol>	

б) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа	
Правильно выполнены все пункты задания. Программа после исправлений для всех натуральных чисел N, не превосходящих 10 <sup>9</sup> , верно определяет максимальную цифру. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в п. 1 задания ответ неверный или отсутствует, или выполнен п. 1, и верно исправлена только одна ошибка). Верное указание на ошибку при её неверном исправлении при этом не засчитывается. 2. Или выполнен п. 1, а вместо указания на ошибки в программе и их исправления приведён новый верный текст решения, возможно, совершенно непохожий на исходный. 3. Или правильно выполнены все действия (приведён верный ответ на вопрос 1, и исправлены обе ошибки), но в текст программы внесены и другие изменения, приводящие к её неверной работе	2
Правильно выполнено только одно действие из трёх, то есть либо только выполнен п. 1, либо он не выполнен или выполнен неверно и верно исправлена только одна ошибка программы путём её явного указания и исправления	1
Все пункты задания выполнены неверно (ответ на п. 1 не приведён или приведён неверно; ошибки не найдены или найдены, но не исправлены или исправлены неверно)	0
<i>Максимальный балл</i>	3

**C2**

Дан целочисленный массив из 20 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 максимальное значение среди трёхзначных элементов массива, не делящихся на 9. Если в исходном массиве нет элемента, значение которого является трёхзначным числом и при этом не кратно 9, то выведите сообщение «Не найдено».

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>Паскаль</b>
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, MAX AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre>const     N = 20; var     a: array [1..N] of integer;     i, j, max: integer; begin     for i := 1 to N do         readln(a[i]);     ... end.</pre>
<b>Си</b>	<b>Алгоритмический язык</b>
<pre>#include &lt;stdio.h&gt; #define N 20 void main() {     int a[N];     int i, j, max;     for (i = 0; i &lt; N; i++)         scanf("%d", &amp;a[i]);     ... }</pre>	<pre><u>алг</u> <u>нач</u>     цел N = 20     целтаб a[1:N]     цел i, j, max     <u>нц</u> для i от 1 до N         <u>ввод</u> a[i]     <u>кц</u>     ... <u>кон</u></pre>
<b>Естественный язык</b>	
<p>Объявляем массив A из 20 элементов. Объявляем целочисленные переменные I, J, MAX. В цикле от 1 до 20 вводим элементы массива A 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 2.4) 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 ответа, не искажающие его смысла)	
<b>На языке Паскаль</b>	
<pre> max := 99; for i := 1 to N do     if (a[i]&gt;=100) and (a[i]&lt;=998) and (a[i] mod 9&lt;&gt;0) and (a[i]&gt;max) then         max := a[i]; if max &gt; 99 then writeln(max) else writeln('Не найдено');</pre>	
<b>На алгоритмическом языке</b>	
<pre> max := 99 нц для i от 1 до N     если a[i]&gt;=100 и a[i]&lt;=998 и mod(a[i],9)&lt;&gt;0 и a[i]&gt;max     то         max := a[i]     все кц если max &gt; 99 то     вывод max иначе     вывод "Не найдено" все</pre>	
<b>На языке Бейсик</b>	
<pre> MAX = 99 FOR I = 1 TO N     IF A(I)&gt;=100 AND A(I)&lt;=999 AND A(I) MOD 9&lt;&gt;0 AND A(I)&gt;MAX THEN         MAX = A(I)     END IF NEXT I IF MAX &gt; 99 THEN     PRINT MAX ELSE     PRINT "Не найдено" END IF</pre>	
<b>На языке Си</b>	
<pre> max = 99; for (i = 0; i&lt;N; i++)     if (a[i]&gt;99 &amp;&amp; a[i]&lt;999 &amp;&amp; a[i]%9!=0 &amp;&amp; a[i]&gt;max)         max = a[i]; if (max&gt;99)     printf("%d", max); else     printf("Не найдено");</pre>	

<b>На естественном языке</b>	
<p>Записываем в переменную MAX начальное значение, равное 99. В цикле от 1-ого элемента до 20-ого находим остаток от деления элемента исходного массива на 9. Если значение данного остатка не равно 0 и значение текущего элемента массива больше 99 и меньше 1000, то сравниваем значение текущего элемента массива со значением переменной MAX. Если текущий элемент массива больше MAX, то записываем в MAX значение этого элемента массива. Переходим к следующему элементу.</p> <p>После завершения цикла проверяем значение переменной MAX. Если оно больше 99, то выводим его, иначе выводим сообщение «Не найдено»</p>	
Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение.</p> <p>Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	2
<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1) не инициализируется или неверно инициализируется переменная MAX (например, присваивается начальное значение, большее или равное 100);</li> <li>2) неточно определяется принадлежность диапазону трёхзначных чисел (например, в сравнении со 100 вместо знака «больше или равно» используется знак «больше»; также обратите внимание, что с учётом ограничения на кратность 9 допустима проверка на принадлежность диапазону от 100 до 998);</li> <li>3) неверно осуществляется проверка делимости на 9;</li> <li>4) на делимость на 9 проверяется не значение элемента, а его индекс;</li> <li>5) в сложном условии вместо логической операции «И» используется логическая операция «ИЛИ»;</li> <li>6) отсутствует вывод ответа, или при выводе ответа не учитывается случай, когда требуемого числа в массиве может не быть;</li> <li>7) используется переменная, не объявленная в разделе описания переменных;</li> <li>8) не указано или неверно указано условие завершения цикла;</li> </ol>	1

9) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;	
10) неверно расставлены операторные скобки	
Ошибок, перечисленных в п. 1–10, две или больше, или алгоритм сформулирован неверно	0
<i>Максимальный балл</i>	2

**С3**

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 **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 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 27. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 27 или больше камней.

В начальный момент в куче было  $S$  камней,  $1 \leq S \leq 26$ .

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.

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 выигрышающий ход для каждого указанного значения  $S$ .

б) Укажите такое значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.

3. Укажите значение  $S$ , при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– 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 рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в куче.

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

1. а) Петя может выиграть, если  $S = 14, \dots, 26$ . Во всех этих случаях достаточно удвоить количество камней. При меньших значениях  $S$  за один ход нельзя получить кучу, в которой больше 26 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет  $S = 13$  камней. Тогда после первого хода Пети в куче будет 14, 15 или 26 камней. Во всех случаях Ваня удваивает количество камней и выигрывает в один ход.

2. Возможные значения  $S$ : 11, 12. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 13 камней: в первом случае добавлением двух камней, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

3. Возможное значение  $S$ : 10. После первого хода Пети в куче будет 11, 12 или 20 камней. Если в куче станет 20 камней, Ваня удвоит количество камней и выигрывает первым ходом. Ситуация, когда в куче 11 или 12 камней, разобрана в п. 2. В этих ситуациях 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).

И.п.	Положения после очередных ходов			
	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>10</b>	$10+1=11$	$11+2=13$	$13+1=14$	<u><math>14*2=28</math></u>
			$13+2=15$	<u><math>15*2=30</math></u>
			$13*2=26$	<u><math>26*2=52</math></u>
	$10+2=12$	$12+1=13$	$13+1=14$	<u><math>14*2=28</math></u>
			$13+2=15$	<u><math>15*2=30</math></u>
			$13*2=26$	<u><math>26*2=52</math></u>
$10*2=20$	<u><math>20*2=40</math></u>			



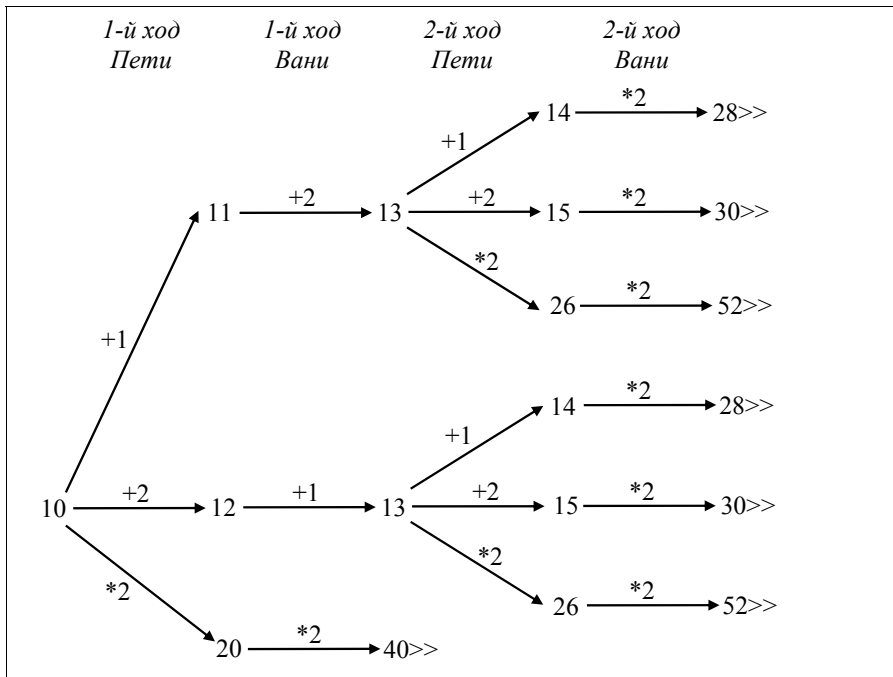


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии. Знаком “>>” обозначены позиции, в которых партия заканчивается

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 полностью, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, то есть показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если: а) правильно указаны все позиции, в которых Петя выигрывает первым ходом; б) правильно указана позиция, в которой Ваня</p>	

<p>выигрывает первым ходом, и явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от выполненного полностью задания состоит в том, что не указаны явно ходы, которыми выигрывает Петя или Ваня.</p> <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. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий стратегии, которую выбрал Ваня.</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. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше)</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> <li>1. Третье задание выполнено полностью.</li> <li>2. Первое и второе задания выполнены полностью.</li> <li>3. Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения <math>S</math></li> </ol>	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> <li>1. Первое или второе задание выполнено полностью.</li> <li>2. 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>3. Первое задание выполнено частично, и для одного из остальных заданий правильно указаны значения <math>S</math>.</li> <li>4. Для второго и третьего заданий правильно указаны значения <math>S</math></li> </ol>	1
<p>Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл</p>	0
<i>Максимальный балл</i>	3

**C4**

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 – наибольшее число  $R$ , удовлетворяющее следующим условиям:

3)  $R$  – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел; допускаются произведения различных элементов последовательности, равных по величине);

4)  $R$  делится на 21.

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 каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение.

*Пример входных данных:*

6  
70  
21  
997  
7  
9  
300  
21000

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

Вычисленное контрольное значение: 21000

Контроль пройден

**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

Произведение двух чисел делится на 21, если:

- один из сомножителей делится на 21 (второй может быть любым) либо
- ни один из сомножителей не делится на 21, причём один из сомножителей делится на 7, а другой – на 3.

Поэтому программа, вычисляющая кодовое число, 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:

$M7$  – самое большое число, кратное 7, но не кратное 3;

$M3$  – самое большое число, кратное 3, но не кратное 7;

$M21$  – самое большое число, кратное 21;

$MAX$  – самое большое число среди всех элементов последовательности, отличное от  $M21$  (если число  $M21$  встретилось более одного раза и оно же является максимальным, то  $MAX = M21$ ).

После того как все данные прочитаны, искомое контрольное значение вычисляется как максимум из произведений  $M21*MAX$  и  $M7*M3$ .

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 M7, M3, M21, R, MAX, dat, res, i, N: longint;
begin
  M7 := 0;
  M3 := 0;
  M21 := 0;
  MAX := 0;
  readln(N);
  for i := 1 to N do
  begin
    readln(dat);
    if ((dat mod 7) = 0) and ((dat mod 3) > 0) and (dat > M7) then
      M7 := dat;
    if ((dat mod 3) = 0) and ((dat mod 7) > 0) and (dat > M3) then
      M3 := dat;
    if (dat mod 21 = 0) and (dat > M21) then
      begin
        if M21 > MAX then MAX := M21;
        M21 := dat;
      end
    else
      if dat > MAX then
        MAX := dat;
  end;
  readln(R);
  if (M7*M3 < M21*MAX) then
    res := M21*MAX;
  else
    res := M7*M3;
  writeln('Вычисленное контрольное значение: ', res);
  if R = res then writeln('Контроль пройден')
    else writeln('Контроль не пройден');
end.
```

Пример правильной и эффективной программы на языке Бейсик:	
<pre> M21 = 0 M7 = 0 M3 = 0 MAX = 0 INPUT N FOR I = 1 TO N   INPUT DAT   IF DAT MOD 7 = 0 AND DAT &gt; M7 THEN     M7 = DAT   ELSE     IF DAT MOD 3 = 0 AND DAT &gt; M3 THEN       M3 = DAT     END IF   END IF   IF DAT MOD 21 = 0 AND DAT &gt; M21 THEN     IF M21 &gt; MAX THEN       MAX = M21     END IF   END IF   M21 = DAT ELSE   IF DAT &gt; MAX THEN     MAX = DAT   END IF END IF NEXT I INPUT R IF M3 * M7 &lt; M21 * MAX THEN   RES = M21 * MAX ELSE   RES = M3 * M7 END IF PRINT "Вычисленное контрольное значение:"; RES IF RES = R THEN   PRINT "Контроль пройден" ELSE   PRINT "Контроль не пройден" END IF END </pre>	
Указания по оцениванию	Баллы
<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</p>	

<p>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 одного из следующих видов:</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 (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку)</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 <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 BPascal или <code>Qbasic</code>) 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 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 входной последовательности).</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. 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.</p> <p>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.</p> <p>Количество синтаксических «описок» не должно быть более семи.</p> <p>Программа может быть неэффективна по времени. Например, все числа запоминаются в массиве (ниже n – количество прочитанных чисел), и перебираются все возможные пары элементов последовательности.</p> <p>Например, так:</p> <pre>max := 0; for i := 1 to n - 1 do for j := i + 1 to n do   if ((a[i]*a[j]) mod 21 = 0) and     ((a[i]*a[j]) &gt; max) then     max := a[i]*a[j];</pre>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.</p> <p>При этом выполнено одно из двух условий.</p> <p>1. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>2. Программа правильно работает в одном из важных частных случаев, например в случае, когда искомое произведение – это произведение наибольшего числа, которое делится на 3, и наибольшего числа, которое делится на 7.</p> <p><u>Допускается любое количество синтаксических ошибок</u></p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p style="text-align: right;"><i>Максимальный балл</i></p>	4