

**Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ****Пояснения к демонстрационному варианту контрольных  
измерительных материалов единого государственного экзамена  
2015 года по ИНФОРМАТИКЕ и ИКТ**

При ознакомлении с демонстрационным вариантом контрольных измерительных материалов ЕГЭ 2015 г. следует иметь в виду, что задания, включённые в него, не отражают всех вопросов содержания, которые будут проверяться с помощью вариантов КИМ в 2015 г. Полный перечень вопросов, которые могут контролироваться на едином государственном экзамене 2015 г., приведён в кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для единого государственного экзамена 2015 г. по информатике и ИКТ.

**В демонстрационном варианте представлено по несколько примеров каждого из заданий 3, 6, 7 и 9. В реальных вариантах экзаменационной работы на каждую позицию будет предложено только одно задание.**

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 и 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»



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

В. Б. Бетелин  
«31» октября 2014 г.

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

**Демонстрационный вариант  
контрольных измерительных материалов единого  
государственного экзамена 2015 года  
по информатике и ИКТ**

подготовлен Федеральным государственным бюджетным  
научным учреждением

«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»



**3** 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.

В каталоге находится 7 файлов:

carga.mp3  
 cascad.mpeg  
 cassa.mp3  
 cassandra.mp4  
 castrol.mp4  
 picasa.map  
 picasa.mp4

Определите, по какой из перечисленных масок из этих 7 файлов будет отобрана указанная группа файлов:

cascad.mpeg  
 cassa.mp3  
 cassandra.mp4  
 picasa.mp4

- 1) \*cas\*a\*.mp\*    2) \*ca\*a\*.mp\*    3) \*cas\*.mp\*    4) \*cas\*a\*.mp?

Ответ:

**ИЛИ**

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 Леоненко В.С.

*Пояснение: дядей считается брат отца или матери.*

ID	Фамилия_И.О.	Пол
14	Леоненко Н.А.	Ж
23	Геладзе И.П.	М
24	Геладзе П.И.	М
25	Геладзе П.П.	М
34	Леоненко А.И.	Ж
35	Леоненко В.С.	Ж
33	Леоненко С.С.	М
42	Вильямс О.С.	Ж
44	Гнейс А.С.	Ж
45	Гнейс В.А.	М
47	Вильямс П.О.	М
57	Паоло А.П.	Ж
64	Моор П.А.	Ж
...	...	...

ID_Родителя	ID_Ребёнка
23	24
44	24
24	25
64	25
23	34
44	34
34	35
33	35
14	33
34	42
33	42
24	57
64	57
...	...

- 1) Геладзе И.П.    2) Геладзе П.И.    3) Гнейс А.С.    4) Леоненко Н.А.

Ответ:

**Ответами к заданиям 4–23 являются число, последовательность букв или цифр, которые следует записать в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишете в отдельной клеточке в соответствии с приведёнными в бланке образцами.**

**4** Сколько единиц в двоичной записи десятичного числа 519?

Ответ: \_\_\_\_\_.

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

	A	B	C	D	E	F	G
A		5		12			25
B	5			8			
C				2	4	5	10
D	12	8	2				
E			4				5
F			5				5
G	25		10		5	5	

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

Ответ: \_\_\_\_\_.

**6**

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

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

Пример. Исходное число: 3165. Суммы:  $3 + 1 = 4$ ;  $6 + 5 = 11$ . Результат: 114.

Укажите наименьшее число, в результате обработки которого, автомат выдаст число 1311.

Ответ: \_\_\_\_\_.

**ИЛИ**

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.

Например, **2121** – это программа

**умножь на 2**

**прибавь 1**

**умножь на 2**

**прибавь 1,**

которая преобразует число 1 в число 7.

Запишите порядок команд в программе преобразования **числа 3 в число 63**, содержащей не более 8 команд, указывая лишь номера команд. Если таких программ более одной, то запишите любую из них.

Ответ: \_\_\_\_\_.

**7** Коле нужно с помощью электронных таблиц построить таблицу квадратов двузначных чисел от 20 до 59.

Для этого сначала в диапазоне B1:K1 он записал числа от 0 до 9, и в диапазоне A2:A5 он записал числа от 2 до 5. Затем в ячейку B5 записал формулу квадрата двузначного числа (A5 – число десятков; B1 – число единиц), после чего скопировал её во все ячейки диапазона B2:K5. В итоге получил таблицу квадратов двузначных чисел. На рисунке ниже представлен фрагмент этой таблицы.

	A	B	C	D	E
1		0	1	2	3
2	2	400	441	484	529
3	3	900	961	1024	1089
4	4	1600	1681	1764	1849
5	5	2500	2601	2704	2809

В ячейке B5 была записана одна из следующих формул:

- 1) =(B1+10\*A5)^2
- 2) =(B1+10\*\$A5)^2
- 3) =(B\$1+10\*\$A5)^2
- 4) =(B1+10\*A\$5)^2

Укажите в ответе номер формулы, которая была записана в ячейке B5  
*Примечание: знак \$ используется для обозначения абсолютной адресации.*

**ИЛИ**

Дан фрагмент электронной таблицы.

	A	B	C
1		4	6
2	=(A1 - 2)/(B1 - 1)	=C1*B1/(4*A1 + 4)	=C1/(A1 - 2)

Какое целое число должно быть записано в ячейке A1, чтобы диаграмма, построенная по значениям ячеек диапазона A2:C2, соответствовала рисунку?

Известно, что все значения ячеек из рассматриваемого диапазона неотрицательны.



Ответ: \_\_\_\_\_.

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

Бейсик	Python
<pre>DIM S, N AS INTEGER S = 33 N = 1 WHILE S &gt; 0     S = S - 7     N = N * 3 WEND PRINT (N)</pre>	<pre>s = 33 n = 1 while s &gt; 0:     s = s - 7     n = n * 3 print(n)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач     цел s, n     s := 33     n := 1     нц пока s &gt; 0         s := s - 7         n := n * 3     кц вывод n кон</pre>	<pre>var s, n: integer; begin     s := 33;     n := 1;     while s &gt; 0 do         begin             s := s - 7;             n := n * 3;         end;     writeln(n) end.</pre>
Си	
<pre>#include&lt;stdio.h&gt; int main(void) {     int s, n;     s = 33;     n = 1;     while (s &gt; 0) {         s = s - 7;         n = n * 3;     }     printf("%d\n", n); }</pre>	

Ответ: \_\_\_\_\_.

9

Производилась двухканальная (стерео) звукозапись с частотой дискретизации 64 кГц и 24-битным разрешением. В результате был получен файл размером 120 Мбайт, сжатие данных не производилось. Определите приблизительно, сколько времени (в минутах) производилась запись. В качестве ответа укажите ближайшее к времени записи целое число, кратное 5.

Ответ: \_\_\_\_\_.

**ИЛИ**

Документ объемом 40 Мбайт можно передать с одного компьютера на другой двумя способами.

А. Сжать архиватором, передать архив по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет  $2^{23}$  бит в секунду;
- объем сжатого архиватором документа равен 90% исходного;
- время, требуемое на сжатие документа, – 16 секунд, на распаковку – 2 секунды?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

*Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.*

Единицы измерения «секунд», «сек.», «с» к ответу добавлять не нужно.

Ответ: \_\_\_\_\_.

10

Все 4-буквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. КККК
2. КККЛ
3. КККР
4. КККТ

.....

Запишите слово, которое стоит под номером **67**.

Ответ: \_\_\_\_\_.

11

Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик	Python
<pre>SUB F(n)   PRINT n   IF n &lt; 5 THEN     F(n + 1)     F(n + 3)   END IF END SUB</pre>	<pre>def F(n):   print(n)   if n &lt; 5:     F(n + 1)     F(n + 3)</pre>
Алгоритмический язык	Паскаль
<pre>алг F(цел n) нач   вывод n, нс   если n &lt; 5 то     F(n + 1)     F(n + 3)   все кон</pre>	<pre>procedure F(n: integer); begin   writeln(n);   if n &lt; 5 then     begin       F(n + 1);       F(n + 3)     end end</pre>
Си	
<pre>void F(int n) {   printf("%d\n", n);   if (n &lt; 5) {     F(n + 1);     F(n + 3);   } }</pre>	

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(1)?

Ответ: \_\_\_\_\_.

- 12** В терминологии сетей TCP/IP маской сети называется 32-разрядная двоичная (то есть состоящая из нулей и единиц) последовательность. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 217.8.244.3  
 Маска: 255.255.252.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	3	8	217	224	244	252	255

*Пример.*

Пусть искомый IP-адрес: 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: *НВАF*

Ответ: \_\_\_\_\_.

- 13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы Ш, К, О, Л, А (таким образом, используется 5 различных символов). Каждый такой пароль в компьютерной системе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Укажите объём памяти в байтах, отводимый этой системой для записи 30 паролей. В ответе запишите только число, слово «байт» писать не нужно.

Ответ: \_\_\_\_\_.

- 14** 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 (буквами *n, a, b* обозначены неизвестные числа, при этом  $n > 1$ ):

НАЧАЛО

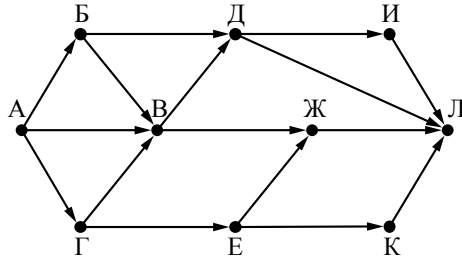
```
    сместиться на (-3, -3)
    ПОВТОРИ n РАЗ
        сместиться на (a, b)
        сместиться на (27, 12)
    КОНЕЦ ПОВТОРИ
    сместиться на (-22, -7)
```

КОНЕЦ

Укажите наименьшее возможное значение числа *n*, для которого найдутся такие значения чисел *a* и *b*, что после выполнения программы Чертёжник возвратится в исходную точку.

Ответ: \_\_\_\_\_.

**15** 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: \_\_\_\_\_.

**16** Сколько единиц содержится в двоичной записи значения выражения:  $4^{2014} + 2^{2015} - 8$ ?

Ответ: \_\_\_\_\_.

**17** 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)
Ухо	35
Подкова	25
Наковальня	40
Ухо   Подкова   Наковальня	70
Ухо & Наковальня	10
Ухо & Подкова	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу Подкова & Наковальня?

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

Ответ: \_\_\_\_\_.

**18** На числовой прямой даны два отрезка:  $P = [37; 60]$  и  $Q = [40; 77]$ . Укажите наименьшую возможную длину такого отрезка А, что формула

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

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

Ответ: \_\_\_\_\_.

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

Определите значение переменной  $c$  после выполнения следующего фрагмента программы, записанного ниже на разных языках программирования.

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

Ответ: \_\_\_\_\_.



**20** Ниже на пяти языках программирования 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 2, а потом 15.

Бейсик	Python
<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>
Алгоритмический язык	Паскаль
<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, 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	
<pre>#include&lt;stdio.h&gt; int main(void) {   int x, a, b;   scanf("%d", &amp;x);   a = 0;   b = 1;   while (x &gt; 0){     a = a + 1;     b = b * (x % 10);     x = x / 10;   }   printf("%d\n%d", a, b); }</pre>	

Ответ: \_\_\_\_\_.

**21** Напишите в ответе число различных значений входной переменной  $k$ , при которых программа выдаёт тот же ответ, что и при входном значении  $k = 64$ . Значение  $k = 64$  также включается в подсчёт различных значений  $k$ . Для Вашего удобства программа приведена на пяти языках программирования.

Бейсик	Python
<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>	<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>
Алгоритмический язык	Паскаль
<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>	<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>
Си	
<pre>#include&lt;stdio.h&gt; int f(int n) {   return n * n; } int main(void) {   int k, i;   scanf("%d", &amp;k);   i = 12;   while (i &gt; 0 &amp;&amp; f(i) &gt;= k)     i--;   printf("%d", i); }</pre>	

Ответ: \_\_\_\_\_.

**22** Исполнитель Май4 преобразует число, записанное на экране. У исполнителя три команды, которым присвоены номера:

1. Прибавь 1
2. Прибавь 2
3. Прибавь 4

Первая из них увеличивает число на экране на 1, вторая увеличивает это число на 2, а третья – на 4. Программа для исполнителя Май4 – это последовательность команд.

Сколько есть программ, которые **число 21** преобразуют в **число 30**?

Ответ: \_\_\_\_\_.

**23** Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8$ , которые удовлетворяют всем перечисленным ниже условиям?

$$\begin{aligned} (x_1 \vee x_2) \wedge ((x_1 \wedge x_2) \rightarrow x_3) \wedge (\neg x_1 \vee y_1) &= 1 \\ (x_2 \vee x_3) \wedge ((x_2 \wedge x_3) \rightarrow x_4) \wedge (\neg x_2 \vee y_2) &= 1 \\ \dots \\ (x_6 \vee x_7) \wedge ((x_6 \wedge x_7) \rightarrow x_8) \wedge (\neg x_6 \vee y_6) &= 1 \\ (x_7 \vee x_8) \wedge (\neg x_7 \vee y_7) &= 1 \\ (\neg x_8 \vee y_8) &= 1 \end{aligned}$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8$ , 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 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

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

Бейсик	Python
<pre> CONST n = 4 count = 0 maximum = 999 FOR I = 1 TO n   INPUT x   IF x mod 2 &lt;&gt; 0 THEN     count = count + 1     IF x &gt; maximum THEN       maximum = I     END IF   END IF NEXT I IF count &gt; 0 THEN   PRINT count   PRINT maximum ELSE   PRINT "NO" END IF                     </pre>	<pre> n = 4 count = 0 maximum = 999 for i in range(1, n + 1):     x = int(input())     if x % 2 != 0:         count += 1         if x &gt; maximum:             maximum = i if count &gt; 0:     print(count)     print(maximum) else:     print("NO")                     </pre>

Алгоритмический язык	Паскаль
<pre> алг нач   цел n = 4   цел i, x   цел maximum, count   count := 0   maximum := 999   нц для i от 1 до n     ввод x     если mod(x, 2) &lt;&gt; 0 то       count := count + 1     если x &gt; maximum то       maximum := i     все   все кц если count &gt; 0 то   вывод count, нс   вывод maximum иначе   вывод "NO" все кон </pre>	<pre> const n = 4; var i, x: integer; var maximum, count: integer; begin   count := 0;   maximum := 999;   for i := 1 to n do   begin     read(x);     if x mod 2 &lt;&gt; 0 then     begin       count := count + 1;       if x &gt; maximum then         maximum := i       end     end;   end;   if count &gt; 0 then   begin     writeln(count);     writeln(maximum)   end   else     writeln('NO')   end. </pre>
<pre> Си #include &lt;stdio.h&gt; int main(void) {   const int n = 4;   int i, x, maximum, count;   count = 0;   maximum = 999;   for (i = 1; i &lt;= n; i++) {     scanf("%d",&amp;x);     if (x % 2 != 0) {       count++;       if (x &gt; maximum)         maximum = i;     }   }   if (count &gt; 0) {     printf("%d\n", count);     printf("%d\n", maximum);   }   else     printf("NO\n"); } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:

2 9 4 3

2. Приведите пример такой последовательности, содержащей хотя бы одно нечётное число, что, 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.

25

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, сумма которых нечётна и положительна. Под парой подразумевается два подряд идущих элемента массива.

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>N = 20 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre>N = 20 i = None j = None k = None a = [int(input()) for i in range(N)] ...</pre>
Алгоритмический язык	Паскаль
<pre>алг нач     цел N = 20     цел таб 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; int main(void) {     const int N = 20;     int a[N];     int i, j, k;     for (i = 0; i &lt; N; i++)         scanf("%d", &amp;a[i]);     ... }</pre>	<p>Объявляем массив A из 20 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й. ...</p>

26

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 35.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 35 или больше камней.  
В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 34$ .

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

- Укажите все такие значения числа  $S$ , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрышающие ходы.
- 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.

27

На спутнике «Фотон» установлен прибор, предназначенный для измерения энергии космических лучей. Каждую минуту прибор передаёт по каналу связи неотрицательное вещественное число – количество энергии, полученной за последнюю минуту, измеренное в условных единицах. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо найти в заданной серии показаний прибора минимальное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Количество энергии, получаемое прибором за минуту, не превышает 1000 условных единиц. Общее количество показаний прибора в серии не превышает 10 000.

*Вам предлагается два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.*

*Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.*

*Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.*

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ А. Максимальная оценка за выполнение задания А равна 2 баллам.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

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.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, равна 4 баллам.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, равна 3 баллам.

**НАПОМИНАЕМ!** Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число  $N$  – общее количество показаний прибора. Гарантируется, что  $N > 6$ . В каждой из следующих  $N$  строк задаётся одно неотрицательное вещественное число – очередное показание прибора.

*Пример входных данных:*

```
11
12
45.3
5.5
4
25
23
21
20
10
12
26
```

Программа должна вывести одно число – описанное в условии произведение.

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

```
48
```

**Система оценивания экзаменационной работы по информатике и ИКТ****Часть 1**

За правильный ответ на задания 1–23 ставится 1 балл; за неверный ответ или его отсутствие – 0 баллов.

№ задания	Ответ	
1	1	
2	2	
3	1	2
4	4	
5	23	
6	2949	21212121
7	3	5
8	243	
9	5	Б14
10	ЛККР	
11	49	
12	DCFA	
13	180	
14	5	
15	13	
16	2013	
17	20	
18	20	
19	6	
20	35	
21	15	
22	96	
23	61	

**Часть 2****Критерии оценивания заданий с развёрнутым ответом****24**

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

Бейсик	Python
<pre> CONST n = 4 count = 0 maximum = 999 FOR I = 1 TO n   INPUT x   IF x mod 2 &lt;&gt; 0 THEN     count = count + 1     IF x &gt; maximum THEN       maximum = I     END IF   END IF NEXT I IF count &gt; 0 THEN   PRINT count   PRINT maximum ELSE   PRINT "NO" END IF </pre>	<pre> n = 4 count = 0 maximum = 999 for i in range(1, n + 1):     x = int(input())     if x % 2 != 0:         count += 1         if x &gt; maximum:             maximum = i if count &gt; 0:     print(count)     print(maximum) else:     print("NO") </pre>

Алгоритмический язык	Паскаль
<pre> алг нач   цел n = 4   цел i, x   цел maximum, count   count := 0   maximum := 999   нц для i от 1 до n     ввод x     если mod(x, 2) &lt;&gt; 0 то       count := count + 1     если x &gt; maximum то       maximum := i     все   все кц если count &gt; 0 то   вывод count, нс   вывод maximum иначе   вывод "NO" все кон </pre>	<pre> const n = 4; var i, x: integer; var maximum, count: integer; begin   count := 0;   maximum := 999;   for i := 1 to n do   begin     read(x);     if x mod 2 &lt;&gt; 0 then     begin       count := count + 1;       if x &gt; maximum then         maximum := i       end     end;   end;   if count &gt; 0 then   begin     writeln(count);     writeln(maximum)   end   else     writeln('NO')   end. </pre>
<pre> Си #include &lt;stdio.h&gt; int main(void) {   const int n = 4;   int i, x, maximum, count;   count = 0;   maximum = 999;   for (i = 1; i &lt;= n; i++) {     scanf("%d",&amp;x);     if (x % 2 != 0) {       count++;       if (x &gt; maximum)         maximum = i;     }   }   if (count &gt; 0) {     printf("%d\n", count);     printf("%d\n", maximum);   }   else     printf("NO\n"); } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:  
2 9 4 3
  2. Приведите пример такой последовательности, содержащей хотя бы одно нечётное число, что, 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)
<p>Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.</p> <ol style="list-style-type: none"> <li>1. Программа выведет два числа: 2 и 999.</li> <li>2. Пример последовательности, содержащей нечётные числа, для которой программа работает правильно: 1 2 3 999. <i>Замечание для проверяющего. В конце работы программы значение переменной maximum всегда равно 999. Соответственно, программа будет работать верно, если в последовательности есть 999. Выведенное количество нечётных чисел будет правильным в любом случае.</i></li> <li>3. В программе есть две ошибки. <b>Первая ошибка:</b> неверная инициализация maximum. Строка с ошибкой: maximum := 999; <b>Верное исправление:</b> maximum := 0; Вместо 0 может быть использовано любое число, меньшее или равное 1. <b>Вторая ошибка:</b> неверное присваивание при вычислении максимума. Строка с ошибкой: maximum = i; <b>Верное исправление:</b> maximum = x;</li> </ol>

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>четыре</b> действия:</p> <p>1) указать, что выведет программа при конкретной входной последовательности;</p> <p>2) указать пример последовательности, при которой программа работает правильно;</p> <p>3) исправить первую ошибку;</p> <p>4) исправить вторую ошибку.</p> <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p> <p>Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия:</p> <p>а) правильно указана строка с ошибкой;</p> <p>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</p>	
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> <p>а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной;</p> <p>б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки</p>	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

25

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, сумма которых нечётна и положительна. Под парой подразумевается два подряд идущих элемента массива.

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>N = 20 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre>N = 20 i = None j = None k = None a = [int(input()) for i in range(N)] ...</pre>
Алгоритмический язык	Паскаль
<pre>алг нач     цел N = 20     цел таб 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; int main(void) {     const int N = 20;     int a[N];     int i, j, k;     for (i = 0; i &lt; N; i++)         scanf("%d", &amp;a[i]);     ... }</pre>	<p>Объявляем массив A из 20 элементов. Объявляем целочисленные переменные I, J, K.</p> <p>В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й.</p> <p>...</p>



<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)
<b>На языке Паскаль</b>
<pre>k := 0; for i := 1 to N - 1 do   if ((a[i]+a[i+1]) mod 2&lt;&gt;0) and (a[i]+a[i+1]&gt;0) then     inc(k); writeln(k);</pre>
<b>На алгоритмическом языке</b>
<pre>k := 0 нц для i от 1 до N - 1   если mod(a[i]+a[i+1],2)&lt;&gt;0 и a[i]+a[i+1]&gt;0     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)+A(I+1)) MOD 2&lt;&gt;0 AND A(I)+A(I+1)&gt;0 THEN     K = K + 1   END IF NEXT I PRINT K</pre>
<b>На языке Python</b>
<pre>k = 0 for i in range(N - 1):   if (a[i] + a[i + 1]) % 2 != 0 and a[i] + a[i + 1] &gt; 0:     k += 1 print(k)</pre>
<b>На языке Си</b>
<pre>k = 0; for (i = 0; i &lt; N - 1; i++)   if ((a[i] + a[i+1]) % 2 != 0 &amp;&amp; a[i] + a[i+1] &gt; 0)     k++; printf("%d", k);</pre>
<b>На естественном языке</b>
<p>Записываем в переменную К начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления суммы текущего и следующего элементов массива на 2. Если значение данного остатка не равно 0 и сумма текущего и следующего элементов массива больше 0, увеличиваем переменную К на единицу. После завершения цикла выводим значение переменной К</p>

<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	2
<p>Не выполнены условия, позволяющие поставить 2 балла. При этом предложено в целом верное решение, содержащее не более одной 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 N);</li> <li>2) не инициализируется или неверно инициализируется счётчик количества найденных пар;</li> <li>3) неверно осуществляется проверка суммы пары элементов на чётность (в том числе используется выражение вида <math>a[i]+a[i+1] \bmod 2 \neq 0</math> без скобок);</li> <li>4) на чётность проверяется не сумма значений элементов, а сумма их индексов;</li> <li>5) неверно осуществляется проверка суммы пары элементов на положительность;</li> <li>6) 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>7) отсутствует вывод ответа;</li> <li>8) используется переменная, не объявленная в разделе описания переменных;</li> <li>9) не указано или неверно указано условие завершения цикла;</li> <li>10) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;</li> <li>11) неверно расставлены операторные скобки.</li> </ol> <p>Допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	1
<p>Не выполнены условия, позволяющие поставить 1 или 2 балла. Например, ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно</p>	0
<i>Максимальный балл</i>	2

**26** 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 35.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 35 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 34$ .

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  $S = 18, \dots, 34$ . При меньших значениях  $S$  за один ход нельзя получить кучу, в которой не менее 35 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 камней 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 один или три камня. В задаче не требуется указать все выигрышные стратегии. Если в работе ученика, как в приведённом примере, просто сказано, 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. 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 выигрывает 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)	
<b>13</b>	$13 + 1 = 14$	$14 + 3 = 17$	$17 + 1 = 18$	$18 * 2 = 36$	
			$17 + 3 = 20$	$20 * 2 = 40$	
			$17 * 2 = 34$	$34 + 1 = 35$	$34 + 3 = 37$
				$34 * 2 = 68$	
			$13 + 3 = 16$	$16 + 1 = 17$	$17 + 1 = 18$
	$17 + 3 = 20$	$20 * 2 = 40$			
	$17 * 2 = 34$	$34 + 1 = 35$			$34 + 3 = 37$
		$34 * 2 = 68$			
	$13 * 2 = 26$	$26 * 2 = 52$			

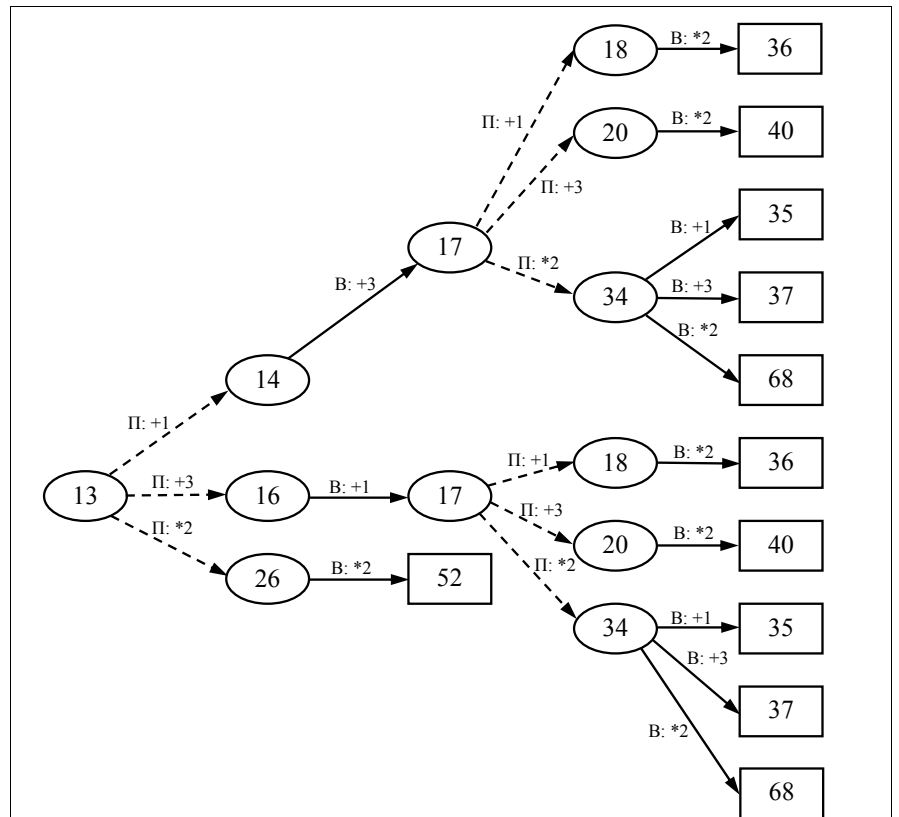


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии. Прямоугольником обозначены позиции, в которых партия заканчивается

*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
В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).  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.  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.

Первое задание считается выполненным частично, если одновременно:

- 1) правильно указаны все позиции, в которых Петя выигрывает первым ходом;
- 2) правильно указана позиция, в которой Ваня выигрывает первым ходом;
- 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.

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.

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.

Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом

Выполнены второе и третье задания. Первое задание выполнено полностью или частично.	3
Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу	
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. <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
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. <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
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл	0
<i>Максимальный балл</i>	3

**27** На спутнике «Фотон» установлен прибор, предназначенный для измерения энергии космических лучей. Каждую минуту прибор передаёт по каналу связи неотрицательное вещественное число – количество энергии, полученной за последнюю минуту, измеренное в условных единицах. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо найти в заданной серии показаний прибора минимальное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Количество энергии, получаемое прибором за минуту, не превышает 1000 условных единиц. Общее количество показаний прибора в серии не превышает 10 000.

*Вам предлагается два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.*

*Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.*

*Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.*

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

**ОБЯЗАТЕЛЬНО** укажите, что программа является решением ЗАДАНИЯ А. Максимальная оценка за выполнение задания А равна 2 баллам.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

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.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

**ОБЯЗАТЕЛЬНО** укажите, что программа является решением ЗАДАНИЯ Б. Максимальная оценка за правильную программу, эффективную по времени и по памяти, равна 4 баллам.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, равна 3 баллам.

**НАПОМИНАЕМ!** Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число  $N$  – общее количество показаний прибора. Гарантируется, что  $N > 6$ . В каждой из следующих  $N$  строк задаётся одно неотрицательное вещественное число – очередное показание прибора.

*Пример входных данных:*

```
11
12
45.3
5.5
4
25
23
21
20
10
12
26
```

Программа должна вывести одно число – описанное в условии произведение.

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

```
48
```

**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание Б (решение для задания А приведено ниже, см. программу 4). Для каждого показания с номером  $k$ , начиная с  $k=7$ , рассмотрим все допустимые по условиям задачи пары, в которых данное показание получено вторым. Минимальное произведение из всех этих пар будет получено, если первым в паре будет взято минимальное показание среди всех, полученных от начала приема и до показания с номером  $k-6$ . Для получения эффективного по времени решения нужно по мере ввода данных помнить минимальное показание на каждый момент времени, каждое вновь полученное показание умножить на минимум, имевшийся на 6 элементов ранее, и выбрать минимальное из всех таких произведений.

Поскольку каждое текущее минимальное показание используется после ввода еще 6 элементов и после этого становится ненужным, достаточно хранить только шесть последних минимумов. Для этого можно использовать массив из 6 элементов и циклически заполнять его по мере ввода данных. Размер этого массива не зависит от общего количества введенных показаний, поэтому такое решение будет эффективным не только по времени, но и по памяти.

Ниже приводится пример такой программы, написанной на алгоритмическом языке.

Программа 1. Пример правильной программы на алгоритмическом языке.

Программа эффективна по времени, и по памяти

```

алг
нач
  цел s = 6 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещ a | очередное показание прибора
  вещтаб мини[0:s-1] | текущие минимумы последних s элементов
  цел i
  | вводим первые s показаний, фиксируем минимумы
  ввод мини[1]
  нц для i от 2 до s
    ввод a
    мини[mod(i, s)] := min(a, мини[i-1])
  кц
  вещ m | минимальное значение произведения
  m := 1000.0 * 1000 + 1
  нц для i от s+1 до N
    ввод a
    m := min(m, a * мини[mod(i, s)])
    мини[mod(i, s)] := min(a, мини[mod(i-1, s)])
  кц
  вывод m
кон

```

Возможны и другие реализации. Например, вместо циклического заполнения массива можно каждый раз сдвигать его элементы. В приведенном ниже примере хранятся и сдвигаются не минимумы, а исходные значения. Решение со сдвигами менее эффективно, чем с циклическим заполнением, но время работы остается пропорциональным  $N$ , поэтому максимальная оценка за такое решение тоже составляет 4 балла.

Программа 2. Пример правильной программы на языке Паскаль.

Программа использует сдвиги, но эффективна по времени и по памяти

```

const s = 6; {требуемое расстояние между показаниями}
var
  N: integer;
  a: array[1..s] of real; {хранение s показаний прибора}
  a_: real; {ввод очередного показания}
  mn: real; {минимальное введенное число не считая s последних}
  m: real; {минимальное значение произведения}
  i, j: integer;
begin
  readln(N);
  {Ввод первых s чисел}
  for i:=1 to s do
    readln(a[i]);
  {Ввод остальных значений, поиск минимального произведения}
  mn := 1001;
  m := 1000 * 1000+1;
  for i := s + 1 to N do begin
    if a[1] < mn then mn := a[1];
    readln(a_);
    if a_ * mn < m then m := a_ * mn;
    {сдвигаем элементы вспомогательного массива влево}
    for j := 1 to s - 1 do
      a[j] := a[j + 1];
    a[s] := a_
  end;
  writeln(m)
end.

```

Если вместо небольшого массива фиксированного размера (циклического или со сдвигами) хранятся все исходные данные (или все текущие минимумы), программа сохраняет эффективность по времени, но становится неэффективной по памяти, так как требуемая память растет пропорционально  $N$ . Ниже приводится пример такой программы на языке Паскаль. Подобная (и аналогичные по сути) программы оцениваются не выше 3 баллов.

<p>Программа 3. Пример правильной программы на Паскале. Программа эффективна по времени, но неэффективна по памяти</p> <pre> const s = 6; {требуемое расстояние между показаниями} var   N: integer;   a: array[1..10000] of real; {хранение всех показаний прибора}   mn: real; {минимальное введенное число}       {не считая s последних}   m: real; {минимальное значение произведения}   i: integer; begin   readln(N);   {Ввод всех показаний прибора}   for i:=1 to N do     readln(a[i]);   mn := 1001;   m := 1000 * 1000 + 1;   for i := s + 1 to N do     begin       if a[i-s] &lt; mn then mn := a[i-s];       if a[i] * mn &lt; m then m := a[i] * mn     end;   writeln(m) end.</pre>
<p>Возможно также переборное решение, в котором находятся произведения всех возможных пар и из них выбирается минимальное. Ниже (см. программу 4) приведен пример подобного решения. Это (и аналогичные ему) решение не эффективно ни по времени, ни по памяти. Оно является решением задачи А, но не является решением задания Б. Оценка за такое решение – 2 балла.</p>
<p>Программа 4. Пример правильной программы на Паскале. Программа не эффективна ни по времени, ни по памяти</p> <pre> const s = 6; {требуемое расстояние между показаниями} var   N: integer;   a: array[1..10000] of real; {хранение всех показаний прибора}   m: real; {минимальное значение произведения}   i, j: integer; begin   readln(N);   {Ввод значений прибора}   for i:=1 to N do     readln(a[i]);   m := 1000 * 1000 + 1;   for i := 1 to N-s do begin     for j := i+s to N do begin       if a[i] * a[j] &lt; m then m := a[i] * a[j];     end;   end;   writeln(m) end.</pre>

Указания по оцениванию	Баллы
<p><i>Предварительные замечания.</i></p> <p>1. В задаче есть два задания (А и Б). Соответственно, ученик может представить две программы. В каждой из программ должно быть указано, решением какого из заданий она является. Если в работе представлена одна программа, то в ней также должно быть указано, решением какого из заданий она является.</p> <p>2. Если ученик не указал, к какому заданию относится программа, или можно предположить, что ученик ошибся в идентификации программ, необходимо следовать приведенным ниже инструкциям.</p> <p>Случай 2.1. Ученик представил только одну программу. Следует рассматривать программу как решение задания Б и оценивать ее по соответствующим критериям.</p> <p>Случай 2.2. Ученик представил две программы, но указание задания есть только для одной из программ. Следует рассматривать вторую программу как ответ на оставшееся задание.</p> <p>Случай 2.3. Ученик представил две программы; ни для одной из них задание не указано или в обоих решениях указано одно и то же задание. Следует первую (по порядку в представленных учеником материалах) программу рассматривать, как ответ на задание А, а вторую – как ответ на задание Б.</p> <p>Случай 2.4. Ученик представил более двух программ. Следует рассматривать только две последние программы и соотносить их с заданиями по правилам 2.1 – 2.3.</p> <p>Случай 2.5. Решение, представленное в качестве решения задания А, по критериям для задания Б может быть оценено в 3 или 4 балла. При этом решение, представленное в качестве решения задания Б, получило меньшую оценку Следует считать, что ученик перепутал имена заданий и оценивать решение, представленное как решение задания А, по критериям задания Б.</p> <p><b>НАПОМИНАЕМ:</b> <i>Итоговый балл за задачу – это больший из баллов, полученных учеником за каждое из двух представленных решений.</i></p>	

<p><i>Пояснения для проверяющих.</i></p> <p>1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то согласно приведенным ниже критериям, его оценка будет такой же, как если бы это решение было представлено в качестве решения задания А.</p> <p>2. Два задания и, соответственно, возможность для экзаменуемого представить две программы дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое дает ему право получить два балла, а затем приступить к поиску более эффективного решения.</p> <p>3. Приведенные в пп. 2.1 – 2.5 правила имеют целью избежать снижения оценки из-за того, что ученик перепутал обозначения заданий. В целом, эти правила эквивалентны такой процедуре проверки:</p> <p><i>Каждая из двух представленных программ оценивается по критериям для программы Б. В качестве итоговой оценки берется наибольшая из двух оценок.</i></p>	
<b>Критерии оценивания задания А</b>	
Программа решает поставленную задачу, записывая массив в память и просматривая все пары элементов массива. Допускается (не в ущерб правильности решения) не рассматривать отдельные пары, например, такие, в которых элементы находятся в массиве ближе порога, указанного в условии задачи. Например, допускается переборное решение, аналогичное вышеприведенной программе 4. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях оценивания задания Б на 3 балла	2
Не выполнены условия, позволяющие поставить 2 балла. Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности.	1
Не выполнены критерии, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл для задания А</i>	2

<b>Критерии оценивания задания Б</b>	
Программа правильно работает для любых соответствующих условию входных данных. При этом не используются массивы и другие структуры данных, размер которых зависит от количества входных элементов, а время работы пропорционально этому количеству. Возможно использование массивов и динамических структур данных (например, контейнеры STL в программе на языке C++) при условии, что в них в каждый момент времени хранится фиксированное число элементов, требующих для хранения меньше 1кб (минимально необходимое количество – шесть; допускается решение с запасом). Программа может содержать не более трёх синтаксических ошибок следующих видов: <ul style="list-style-type: none"> <li>– пропущен или неверно указан знак пунктуации (запятая, точка с запятой, скобки и т.д.);</li> <li>– неверно написано или пропущено служебное слово языка программирования;</li> <li>– не описана или неверно описана переменная;</li> <li>– применяется операция, недопустимая для соответствующего типа данных.</li> </ul> К синтаксическим ошибкам приравнивается использование неверного типа данных (например, использование целого типа вместо вещественного для представления данных при вводе и обработке). Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку	4
Не выполнены условия, позволяющие поставить 4 балла. Программа правильно работает для любых соответствующих условию входных данных, время работы пропорционально количеству входных элементов. Размер используемой памяти не имеет значения и может зависеть от объема входных данных. В частности, допускается использование одного или нескольких массивов размера $N$ (как в вышеприведенной программе 3). Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла. Кроме того, допускается наличие не более одной содержательной ошибки из числа следующих: <ul style="list-style-type: none"> <li>– неверная инициализация при поиске минимального значения;</li> <li>– неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при <math>6 &lt; N &lt; 12</math>;</li> <li>– неточное определение границ массива, выход за границу массива (например, описан массив с границами от 1 до 6, а реально используется от 0 до 5 или наоборот);</li> </ul>	3



<ul style="list-style-type: none"> <li>– вычисленный индекс элемента массива на 1 отличается от верного;</li> <li>– используется операция <math>&lt;</math> вместо <math>&lt;=</math>, <math>or</math> вместо <math>and</math> и т.п.</li> </ul>	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла.                  Программа работает в целом верно, эффективно или нет. Например, допускается переборное решение, аналогичное вышеприведенной программе 4.                  Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии на 4 балла).                  Допускается до двух содержательных ошибок, описанных в критериях на 3 балла</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.                  Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. Заметим, что неверным является решение, основанное на следующих рассуждениях. Найдем 7 минимальных элементов, используя вспомогательный массив из 7 значений, запоминая еще и индексы найденных элементов. Тогда среди них обязательно найдутся два, находящиеся на расстоянии не меньше 6. Среди таких пар и выберем ответ. Это неверно, так как ответ может выражаться через одно из этих 7 минимальных значений и находящееся именно от него на расстоянии не меньше 6 число, не попавшее в 7 минимальных. Например, 10 5 10 10 10 10 11. В данном случае описанный алгоритм выдаст 100, а правильный ответ 55. Тем не менее, за такое решение можно начислить 1 балл, так как в ряде случаев с его помощью можно получить верный ответ. То же касается других решений, верных лишь в частных случаях.</p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<i>Максимальный балл для задания Б</i>	4
<i>Итоговый максимальный балл</i>	4