

ПРОЕКТ

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»

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.

Демонстрационный вариант
контрольных измерительных материалов
для проведения в 2015 году единого государственного экзамена по
ИНФОРМАТИКЕ и ИКТ

Инструкция по выполнению работы

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

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

Ответы к заданиям 1, 2 и 3 записываются в виде одной цифры, которая соответствует номеру правильного ответа. Эту цифру запишите в бланк ответов № 1.

КИМ

Ответ: 4 [2 4] Бланк

Ответы к заданиям 4–23 записываются в виде числа, последовательности букв или цифр. Ответ запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.

КИМ

Ответ: ЛККР [10 Л К К Р] Бланк

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

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

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

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

Желаем успеха!

Часть 1

Ответом к заданиям 1–3 является одна цифра, которая соответствует номеру правильного ответа. Запишите эту цифру в поле ответа в тексте работы, а затем перенесите в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки.

1

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А – 0; Б – 100; В – 1010; Г – 111; Д – 110. Требуется сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно. Коды остальных букв меняться не должны. Каким из указанных способов это можно сделать?

- 1) для буквы В – 101
- 2) это невозможно
- 3) для буквы В – 010
- 4) для буквы Б – 10

Ответ: []

2

Александра заполняла таблицу истинности для выражения F. Она успела заполнить лишь небольшой фрагмент таблицы:

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

Каким выражением может быть F?

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

Ответ: []

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 > 0 S = S - 7 N = N * 3 WEND PRINT (N)</pre>	<pre>s = 33 n = 1 while s > 0: s = s - 7 n = n * 3 print(n)</pre>
Паскаль	Алгоритмический язык
<pre>var s, n: integer; begin s := 33; n := 1; while s > 0 do begin s := s - 7; n := n * 3; end; writeln(n) end.</pre>	<pre>алг нач цел s, n s := 33 n := 1 нц пока s > 0 s := s - 7 n := n * 3 кц вывод n кон</pre>
Си	
<pre>#include<stdio.h> int main(void) { int s, n; s = 33; n = 1; while (s > 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 < 5 THEN F(n + 1) F(n + 3) END IF END SUB</pre>	<pre>def F(n): print(n) if n < 5: F(n + 1) F(n + 3)</pre>
Паскаль	Алгоритмический язык
<pre>procedure F(n: integer); begin writeln(n); if n < 5 then begin F(n + 1); F(n + 3) end end</pre>	<pre>алг F(цел n) нач вывод n, нс если n < 5 то F(n + 1) F(n + 3) все кон</pre>
Си	
<pre>void F(int n) { printf("%d\n", n); if (n < 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

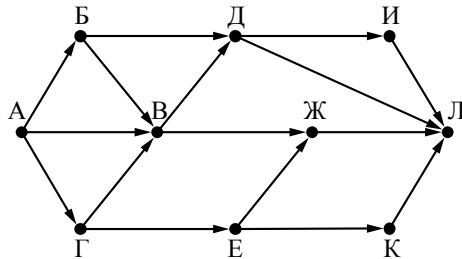
```
    сместиться на (-3, -3)
    ПОВТОРИ n РАЗ
        сместиться на (a, b)
        сместиться на (27, 12)
    КОНЕЦ ПОВТОРИ
    сместиться на (-22, -7)
```

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) < A(i) THEN c = c + 1 t = A(i) A(i) = A(i - 1) A(i - 1) = t ENDF NEXT i </pre>	<pre> c = 0 for i in range(1, 10): if A[i - 1] < A[i]: c = c + 1 t = A[i] A[i] = A[i - 1] A[i - 1] = t </pre>
Паскаль	Алгоритмический язык
<pre> c := 0; for i := 1 to 9 do if A[i - 1] < A[i] then begin c := c + 1; t := A[i]; A[i] := A[i - 1]; A[i - 1] := t; end; </pre>	<pre> c := 0 нц для i от 1 до 9 если A[i - 1] < A[i] то c := c + 1 t := A[i] A[i] := A[i - 1] A[i - 1] := t все кц </pre>
Си	
<pre> c = 0; for (i = 1; i <= 9; i++) if (A[i - 1] < A[i]) { c++; 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 > 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 > 0: a = a + 1 b = b * (x % 10) x = x // 10 print(a) print(b)</pre>
Паскаль	Алгоритмический язык
<pre>var x, a, b: integer; begin readln(x); a:=0; b:=1; while x>0 do begin a:=a+1; b:=b*(x mod 10); x:= x div 10 end; writeln(a); write(b) end.</pre>	<pre>алг нач цел x, a, b ввод x a:=0; b:=1 нц пока x>0 a:=a+1 b:=b*mod(x,10) x:=div(x,10) кц вывод a, нс, b кон</pre>
Си	
<pre>#include<stdio.h> int main(void) { int x, a, b; scanf("%d", &x); a = 0; b = 1; while (x > 0){ a = a + 1; b = b * (x % 10); x = x / 10; } printf("%d\n%d", a, b); }</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 > 0 AND F(I) >= 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 > 0 and f(i) >= k: i = i - 1 print(i)</pre>
Паскаль	Алгоритмический язык
<pre>var k, i : longint; function f(n: longint) : longint; begin f := n * n end; begin readln(k); i := 12; while (i>0) and (f(i)>=k) do i := i-1; writeln(i) end.</pre>	<pre>алг нач цел i, k ввод k i := 12 нц пока i > 0 и f(i) >= k i := i - 1 кц вывод i кон алг цел f(цел n) нач знач := n * n кон</pre>
Си	
<pre>#include<stdio.h> int f(int n) { return n * n; } int main(void) { int k, i; scanf("%d", &k); i = 12; while (i > 0 && f(i) >= 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?

$$(x_1 \vee x_2) \wedge ((x_1 \wedge x_2) \rightarrow x_3) \wedge (\neg x_1 \vee y_1) = 1$$

$$(x_2 \vee x_3) \wedge ((x_2 \wedge x_3) \rightarrow x_4) \wedge (\neg x_2 \vee y_2) = 1$$

...

$$(x_6 \vee x_7) \wedge ((x_6 \wedge x_7) \rightarrow x_8) \wedge (\neg x_6 \vee y_6) = 1$$

$$(x_7 \vee x_8) \wedge (\neg x_7 \vee y_7) = 1$$

$$(\neg x_8 \vee y_8) = 1$$

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 <> 0 THEN count = count + 1 IF x > maximum THEN maximum = I END IF END IF NEXT I IF count > 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 > maximum: maximum = i if count > 0: print(count) print(maximum) else: print("NO") </pre>

Паскаль	Алгоритмический язык
<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 <> 0 then begin count := count + 1; if x > maximum then maximum := i end end; end; if count > 0 then begin writeln(count); writeln(maximum) end else writeln('NO') end. end.</pre>	<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) <> 0 то count := count + 1 если x > maximum то maximum := i все все кц если count > 0 то вывод count, нс вывод maximum иначе вывод "NO" все кон</pre>
<p>Си</p> <pre>#include <stdio.h> int main(void) { const int n = 4; int i, x, maximum, count; count = 0; maximum = 999; for (i = 1; i <= n; i++) { scanf("%d",&x); if (x % 2 != 0) { count++; if (x > maximum) maximum = i; } } if (count > 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>
Паскаль	Алгоритмический язык
<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>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 20 <u>цел таб</u> a[1:N] <u>цел</u> i, j, k <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Си	Естественный язык
<pre>#include <stdio.h> int main(void) { const int N = 20; int a[N]; int i, j, k; for (i = 0; i < N; i++) scanf("%d", &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.

Напишите на любом языке программирования программу для решения поставленной задачи. Оценка выполнения задания будет зависеть не только от правильности программы, но и от того, насколько она эффективна.

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.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

Максимальная оценка за правильную программу, неэффективную и по времени, и по памяти, – 2 балла.

Перед программой укажите версию языка и кратко опишите использованный алгоритм.

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
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	3
8	243	
9	5	Б14
10	ЛККР	
11	49	
12	DCFA	
13	180	
14	5	
15	13	
16	2013	
17	20	
18	17	
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 <> 0 THEN count = count + 1 IF x > maximum THEN maximum = I END IF END IF NEXT I IF count > 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 > maximum: maximum = i if count > 0: print(count) print(maximum) else: print("NO")</pre>

Паскаль	Алгоритмический язык
<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 <> 0 then begin count := count + 1; if x > maximum then maximum := i end end; if count > 0 then begin writeln(count); writeln(maximum) end else writeln('NO') end. end.</pre>	<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) <> 0 то count := count + 1 если x > maximum то maximum := i все все кц если count > 0 то вывод count, нс вывод maximum иначе вывод "NO" все кон</pre>
<pre>Си #include <stdio.h> int main(void) { const int n = 4; int i, x, maximum, count; count = 0; maximum = 999; for (i = 1; i <= n; i++) { scanf("%d",&x); if (x % 2 != 0) { count++; if (x > maximum) maximum = i; } } if (count > 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)

Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.

1. Программа выведет два числа: 2 и 999.
2. Пример последовательности, содержащей нечётные числа, для которой программа работает правильно: 1 2 3 999.

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.

3. В программе есть две ошибки.

Первая ошибка: неверная инициализация maximum.

Строка с ошибкой:

maximum := 999;

Верное исправление:

maximum := 0;

Вместо 0 может быть использовано любое число, меньшее или равное 1.

Вторая ошибка: неверное присваивание при вычислении максимума.

Строка с ошибкой:

maximum = i;

Верное исправление:

maximum = x;

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <ol style="list-style-type: none"> 1) указать, что выведет программа при конкретной входной последовательности; 2) указать пример последовательности, при которой программа работает правильно; 3) исправить первую ошибку; 4) исправить вторую ошибку. <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p> <p>Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия:</p> <ol style="list-style-type: none"> а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа 	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <ol style="list-style-type: none"> а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки 	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из 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 ...</pre>	<pre>N = 20 i = None j = None k = None a = [int(input()) for i in range(N)] ...</pre>
Паскаль	Алгоритмический язык
<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>	<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 ...</pre>
Си	Естественный язык
<pre>#include <stdio.h> int main(void) { const int N = 20; int a[N]; int i, j, k; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... }</pre>	<p>Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й. ...</p>

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
На языке Паскаль
<pre>k := 0; for i := 1 to N - 1 do if ((a[i]+a[i+1]) mod 2 <> 0) and (a[i]+a[i+1] > 0) then inc(k); writeln(k);</pre>
На алгоритмическом языке
<pre>к := 0 нц для i от 1 до N - 1 если mod(a[i]+a[i+1], 2) <> 0 и a[i]+a[i+1] > 0 то к := к + 1 все кц вывод к</pre>
На языке Бейсик
<pre>K = 0 FOR I = 1 TO N - 1 IF (A(I)+A(I+1)) MOD 2 <> 0 AND A(I)+A(I+1) > 0 THEN K = K + 1 END IF NEXT I PRINT K</pre>
На языке Python
<pre>k = 0 for i in range(N - 1): if (a[i] + a[i + 1]) % 2 != 0 and a[i] + a[i + 1] > 0: k += 1 print(k)</pre>
На языке Си
<pre>k = 0; for (i = 0; i < N - 1; i++) if ((a[i] + a[i+1]) % 2 != 0 && a[i] + a[i+1] > 0) k++; printf("%d", k);</pre>
На естественном языке
<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>

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы	2
Не выполнены условия, позволяющие поставить 2 балла. При этом предложено в целом верное решение, содержащее не более одной ошибки из числа следующих: 1) в цикле происходит выход за границу массива (например, используется цикл от 1 до N); 2) не инициализируется или неверно инициализируется счётчик количества найденных пар; 3) неверно осуществляется проверка суммы пары элементов на чётность (в том числе используется выражение вида $a[i]+a[i+1] \bmod 2 < 0$ без скобок); 4) на чётность проверяется не сумма значений элементов, а сумма их индексов; 5) неверно осуществляется проверка суммы пары элементов на положительность; 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); 7) отсутствует вывод ответа; 8) используется переменная, не объявленная в разделе описания переменных; 9) не указано или неверно указано условие завершения цикла; 10) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 11) неверно расставлены операторные скобки. Допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы	1
Не выполнены условия, позволяющие поставить 1 или 2 балла. Например, ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно	0
Максимальный балл	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 выигрышающие 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)
13	$13 + 1 = 14$	$14 + 3 = 17$	$17 + 1 = 18$	<u>$18 * 2 = 36$</u>
			$17 + 3 = 20$	<u>$20 * 2 = 40$</u>
			$17 * 2 = 34$	<u>$34 + 1 = 35$</u>
				<u>$34 + 3 = 37$</u>
				<u>$34 * 2 = 68$</u>
			$13 + 3 = 16$	$16 + 1 = 17$
	$17 + 3 = 20$	<u>$20 * 2 = 40$</u>		
	$17 * 2 = 34$	<u>$34 + 1 = 35$</u>		
		<u>$34 + 3 = 37$</u>		
	$13 * 2 = 26$	<u>$26 * 2 = 52$</u>		

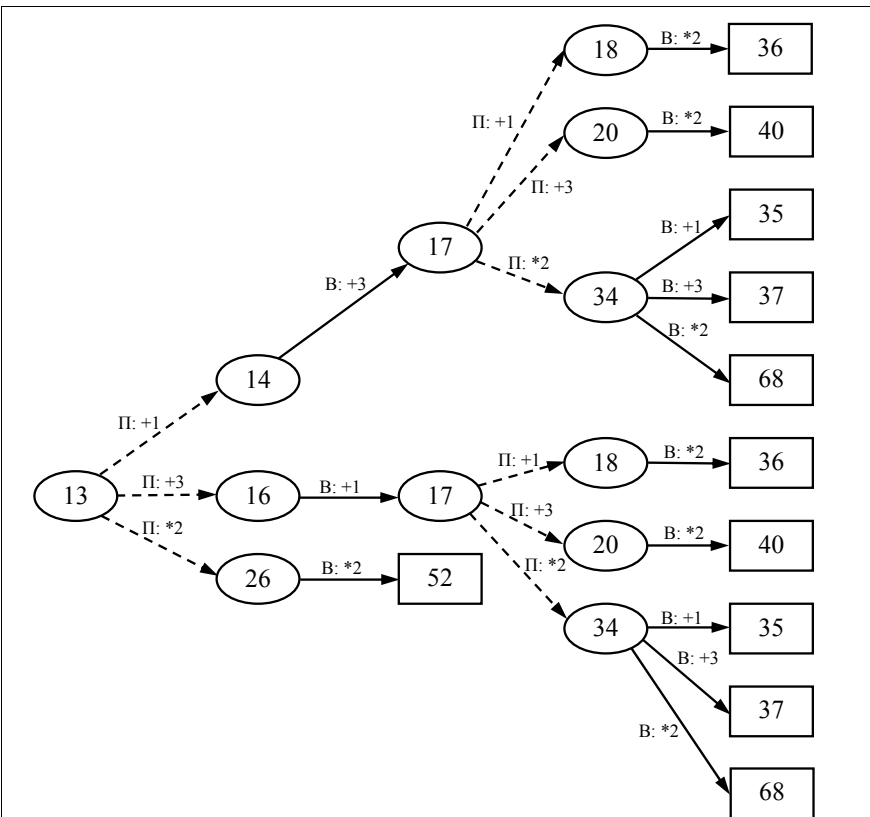


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии.

Прямоугольником обозначены позиции, в которых партия заканчивается

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

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

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

Напишите на любом языке программирования программу для решения поставленной задачи. Ваша оценка будет зависеть не только от правильности программы, но и от того, насколько она эффективна. 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.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

Максимальная оценка за правильную программу, неэффективную ни по времени, ни по памяти, – 2 балла.

Перед программой укажите версию языка и кратко опишите использованный алгоритм.

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
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)

Для построения программы, эффективной по времени, можно определить для каждого элемента входных данных минимальное значение от начала данных до этого элемента включительно. Затем нужно умножить каждый элемент, начиная с седьмого, на значение этого минимума, взятого на шесть элементов раньше, и выбрать наименьшее из этих произведений. Предложенный алгоритм реализован в следующей программе на алгоритмическом языке

**Программа 1а. Пример правильной программы на алгоритмическом языке.
Программа эффективна по времени, но неэффективна по памяти**

```

алг
нач
  цел s = 6 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещтаб a[1:N] | все показания прибора
  цел i
  нц для i от 1 до N
    ввод a[i]
  кц
  вещтаб мини[1:N] | мини[i] - минимум из
                  | первых i показаний

  мини[1] := a[1]
  нц для i от 2 до N
    мини[i] := min(мини[i-1], a[i])
  кц
  вещ m | минимальное значение произведения
  m := 1.0 * 1000 * 1000 + 1
  нц для i от s + 1 до N
    m := min(m, a[i] * мини[i - s])
  кц
  вывод m
кон
    
```

Можно вместо троекратного обращения к каждому элементу делать все необходимые операции с ним сразу после чтения. В этом случае нужен только массив для хранения минимумов, а сами элементы можно не хранить. Этот путь реализован в следующей программе

**Программа 1б. Пример правильной программы на алгоритмическом языке.
Программа эффективна по времени, но неэффективна по памяти**

```

алг
нач
  цел s = 6 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещ a | очередное показание прибора
  вещтаб мини[1:N] | мини[i] - минимум из
                  | первых i показаний

  цел i
  | ввод первых показаний, допустимых пар пока нет
  ввод мини[1]
  нц для i от 2 до s
    ввод a
    мини[i] := min(a, мини[i - 1])
  кц
  | ввод остальных показаний, проверка пар
  вещ m | минимальное значение произведения
  m := 1.0 * 1000 * 1000 + 1
  нц для i от s + 1 до N
    ввод a
    мини[i] := min(a, мини[i - 1])
    m := min(m, a * мини[i - s])
  кц
  вывод m
кон
    
```

На рисунках 1в и 1г приведены примеры, эффективных по времени, но неэффективных по памяти программ, написанных на языках Паскаль и Python.

**Программа 1в. Пример правильной программы на языке Python.
Программа эффективна по времени, но неэффективна по памяти**

```

s = 6
result = 1000 * 1000
N = int(input())
prev_min = [0] * N
prev_min[0] = float(input())
for i in range(1, s):
    prev_min[i] = min(float(input()), prev_min[i - 1])
for i in range(s, N):
    next_num = float(input())
    result = min(result, next_num * prev_min[i - s])
    prev_min[i] = min(prev_min[i - 1], next_num)
print(result)
    
```

Программа 1г. Пример правильной программы на языке Паскаль.
Программа эффективна по времени, но неэффективна по памяти

```
program c4;

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 6 последних}
  m: real; {минимальное значение произведения}
  i: integer;
begin
  readln(N);
  {Ввод значений 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] < mn then mn := a[i - s];
    if a[i] * mn < m then m := a[i] * mn;
  end;
  writeln(m);
end.
```

Все приведённые программы эффективны по времени, но неэффективны по памяти: используемая память пропорциональна объёму исходных данных. Такие (и аналогичные по сути) программы оцениваются не выше 3 баллов. Чтобы построить программу, эффективную по памяти, заметим, что, поскольку при обработке очередного элемента входных данных используется минимум, найденный на шесть элементов раньше, достаточно хранить только шесть последних минимумов. Весь алгоритм содержательно остаётся тем же, но нужно аккуратно организовать работу с массивом минимумов из шести элементов. Ниже приводятся примеры программы на различных языках, реализующей эту идею.

Программа 2а. Пример правильной программы на алгоритмическом языке.
Программа эффективна и 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 6 элементов

  цел i
  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 := 1.0 * 1000 * 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 этой идеи. Один из таких примеров приведён ниже. В этом случае в массиве длины 6 хранятся не минимумы, а исходные значения

Программа 2б. Ещё один пример правильной программы на алгоритмическом языке. Программа эффективна и по времени, и по памяти

```
алг
нач
  цел s = 6 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещтаб a[0:s - 1] | k-е введенное число
      | записываем в ячейку a[mod(k, 6)]
  вещ a_ | очередное показание прибора
  цел i
  | Пролог. Ввод первых шести чисел
  нц для i от 1 до s
    ввод a_
    a[mod(i, s)] := a_
  кц
  | Ввод остальных значений,
  | поиск минимального произведения
  вещ мини = 1001 | минимальное введенное число
      | (не считая 6 последних)
  m := 1.0 * 1000 * 1000 + 1
  нц для i от s + 1 до N
    ввод a_
    мини := min(мини, a[mod(i, s)])
    m := min(m, a_ * мини)
    a[mod(i, s)] := a_
  кц
  вывод m
кон
```

Далее приведены аналогичные программы на языках Python и Паскаль. В решении на языке Python используется «зацикленный» список prev_min из s = 6 элементов, в котором хранятся последовательные минимумы считанных чисел. Если программа получает на вход последовательность чисел a[0], a[1], ..., a[N-1], то сначала заполняются элементы списка prev_min:

$prev_min[0]=a[0]$, $prev_min[1] = \min(a[0], a[1])$, $prev_min[2] = \min(a[0], a[1], a[2])$, ... , $prev_min[s - 1] = \min(a[0], a[1], \dots, a[s - 1])$. При этом допустимых пар элементов (расстояние между которыми не меньше s) нет, поэтому значение переменной $result$, в которой хранится ответ, не обновляется.

Далее считывается значение элемента последовательности $a[s]$, рассматривается его произведение на $prev_min[0] = a[0]$, при необходимости (если полученное произведение меньше значения $result$) обновляется значение $result$. После этого в элемент списка $prev_min[0]$ записывается минимум из считанного числа $a[s]$ и $prev_min[s - 1]$, тем самым $prev_min[0]$ становится равным $\min(a[0], a[1], \dots, a[s])$.

Далее, на каждом шаге в переменную $next_num$ считывается очередной элемент последовательности $a[i]$, он перемножается с $prev_min[i \% s]$, в котором в тот момент хранится $\min(a[0], a[1], \dots, a[i - s])$, при необходимости обновляется переменная $result$, и далее в результате выполнения присваивания $prev_min[i \% s] = \min(prev_min[(i - 1) \% s], next_num)$ значение $prev_min[i \% s]$ становится равно $\min(a[0], a[1], \dots, a[i])$. Это значение $prev_min[i \% s]$ будет затем использовано через s шагов выполнения внешнего цикла, т.е. когда будет считан элемент $a[i + s]$.

При этом все элементы считанной последовательности не сохраняются в списке

Программа 2в. Пример правильной программы на языке Python.

Программа эффективна и по времени, и по памяти

```
s = 6
result = 1000 * 1000
N = int(input())
prev_min = [0] * s
prev_min[0] = float(input())
for i in range(1, s):
    prev_min[i] = min(float(input()), prev_min[i - 1])
for i in range(s, N):
    next_num = float(input())
    result = min(result, next_num * prev_min[i % s])
    prev_min[i % s] = min(prev_min[(i - 1) % s], next_num)
print(result)
```

В программе на языке Паскаль, в отличие от программ 2а - 2в, на обработку очередного числа тратится время, пропорциональное величине задержки (в данной задаче – 6). Это решение менее эффективно, чем решения, приведенные выше. Однако по условию задачи оно также оценивается в 4 балла, так как время обработки очередного числа не зависит от числа введенных чисел.

Программа 2г. Пример правильной программы на языке Паскаль.
Программа эффективна и по времени, и по памяти

```
program c4;
const s = 6; {требуемое расстояние между показаниями}
var
  N: integer;
  a: array[0..s - 1] of real; {хранение показаний прибора}
  {k-е введенное число записываем в ячейку a[k mod 6]}
  a_: real; {ввод очередного показания}
  mn: real; {минимальное введенное число}
  {не считая 6 последних}
  m: real; {минимальное значение произведения}
  i: integer;
begin
  readln(N);
  { Пролог. Ввод первых шести чисел}
  for i:=1 to s do
  begin
    readln(a_);
    a[i mod s] := a_
  end;
  {Ввод остальных значений, поиск минимального произведения}
  mn := 1001; m := 1000 * 1000+1;
  for i := s + 1 to N do
  begin
    readln(a_);
    if a[i mod s] < mn then mn := a[i mod s];
    if a_ * mn < m then m := a_ * mn;
    a[i mod s] := a_
  end;
  writeln(m)
end.
```

В заключение приведем примеры программ на языках Паскаль и Python, которые неэффективны ни по времени, ни по памяти. Подобные программы при отсутствии ошибок получают оценку 2 балла

Программа 3а. Пример правильной программы на языке Python.

Программа неэффективна ни по времени, ни по памяти

```
s = 6
result = 1000 * 1000
N = int(input())
elems = [0] * N
for i in range(N):
    elems[i] = float(input())
for i in range(N):
    for j in range(i + s, N):
        result = min(result, elems[i] * elems[j])
print(result)
```

<p>Программа 3б. Пример правильной программы на языке Паскаль. Программа неэффективна ни по времени, ни по памяти</p> <pre> program c4; 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 for j := i + s to N do if a[i] * a[j] < m then m := a[i] * a[j]; writeln(m); end.</pre>	
Указания по оцениванию	Баллы
<p>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 не более 15 элементов (минимально необходимое количество – шесть; допускается решение с запасом).</p> <p>Программа может содержать не более трёх синтаксических ошибок следующих видов:</p> <ul style="list-style-type: none"> – пропущен или неверно указан знак пунктуации (запятая, точка с запятой, скобки и т.д.); – неверно написано или пропущено служебное слово языка программирования; – не описана или неверно описана переменная; – применяется операция, не допустимая для соответствующего типа данных. <p>К синтаксическим ошибкам приравнивается использование неверного типа данных (например, использование целого типа вместо вещественного для представления данных при вводе и обработке).</p> <p>Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку</p> <p>См. примеры 2а – 2г.</p>	4

<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>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 объёма входных данных. В частности, допускается использование одного или нескольких массивов размера N (как в первых двух приведённых выше решениях).</p> <p>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:</p> <ul style="list-style-type: none"> – неверная инициализация при поиске минимального значения; – неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при $6 < N < 12$; – 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); – вычисленный индекс элемента массива на 1 отличается от верного; – используется знак “<” вместо “<=”, “or” вместо “and” и т.п. <p>См. примеры 1а – 1г.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла.</p> <p>Программа работает в целом верно, эффективно или нет. Например, допускается решение, в котором все элементы хранятся в массиве и производится перебор всех пар, расстояние между которыми не меньше 6.</p> <p>Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии на 4 балла).</p> <p>Допускается до двух «содержательных» ошибок, описанных в критериях на 3 балла</p> <p>См. примеры 3а, 3б.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.</p> <p>Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может отсутствовать или быть представленной отдельными фрагментами, без ограничений на количество ошибок</p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<i>Максимальный балл</i>	4

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

Кодификатор
элементов содержания и требований к уровню
подготовки выпускников образовательных
организаций для проведения
единого государственного экзамена
по информатике и ИКТ

подготовлен Федеральным государственным бюджетным
научным учреждением

«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»

Кодификатор
элементов содержания и требований к уровню подготовки выпускников
образовательных организаций для проведения единого государственного
экзамена по ИНФОРМАТИКЕ и ИКТ

Кодификатор элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения единого государственного экзамена по информатике и ИКТ (далее – кодификатор) является одним из документов, определяющих структуру и содержание КИМ ЕГЭ. Он составлен на основе Федерального компонента государственных стандартов основного общего и среднего (полного) общего образования по информатике и ИКТ (базовый и профильный уровни) (приказ Минобрнауки России от 05.03.2004 № 1089).

В кодификатор не включены элементы содержания, выделенные курсивом в разделе стандарта «Обязательный минимум содержания основных образовательных программ»: данное содержание подлежит изучению, но не включено в раздел стандарта «Требования к уровню подготовки выпускников», т.е. не является объектом контроля. Также в кодификатор не включены те требования к уровню подготовки выпускников, достижение которых не может быть проверено в рамках единого государственного экзамена.

Раздел 1. Перечень элементов содержания, проверяемых на едином государственном экзамене по информатике и ИКТ

Перечень элементов содержания, проверяемых на едином государственном экзамене по информатике и ИКТ, составлен на основе раздела «Обязательный минимум содержания основных образовательных программ» Федерального компонента государственных стандартов среднего (полного) общего образования по информатике и ИКТ (базовый и профильный уровни)

В первом столбце указан код раздела, которому соответствуют крупные блоки содержания. Во втором столбце приводится код элемента содержания, для которого создаются проверочные задания. В третьем столбце приводится словесное описание контролируемого элемента содержания.

Код раздела	Код контролируемого элемента	Элементы содержания, проверяемые на ЕГЭ
1		Информация и информационные процессы
	1.1	Информация и ее кодирование
	1.1.1	Виды информационных процессов
	1.1.2	Процесс передачи информации, источник и приемник

		информации. Сигнал, кодирование и декодирование. Искажение информации
1.1.3		Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации. Единицы измерения количества информации
1.1.4		Скорость передачи информации
1.2		Системы, компоненты, состояние и взаимодействие компонентов. Информационное взаимодействие в системе, управление, обратная связь
1.3		Моделирование
1.3.1		Описание (информационная модель) реального объекта и процесса, соответствие описания объекту и целям описания. Схемы, таблицы, графики, формулы как описания
1.3.2		Математические модели
1.3.3		Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности
1.4		Системы счисления
1.4.1		Позиционные системы счисления
1.4.2		Двоичное представление информации
1.5		Логика и алгоритмы
1.5.1		Высказывания, логические операции, кванторы, истинность высказывания
1.5.2		Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности
1.5.3		Индуктивное определение объектов
1.5.4		Вычислимые функции, полнота формализации понятия вычислимости, универсальная вычислимая функция
1.5.5		Кодирование с исправлением ошибок
1.5.6		Сортировка
1.6		Элементы теории алгоритмов
1.6.1		Формализация понятия алгоритма
1.6.2		Вычислимость. Эквивалентность алгоритмических моделей
1.6.3		Построение алгоритмов и практические вычисления
1.7		Языки программирования
1.7.1		Типы данных
1.7.2		Основные конструкции языка программирования. Система программирования
1.7.3		Основные этапы разработки программ. Разбиение задачи на подзадачи

2		Информационная деятельность человека
	2.1	Профессиональная информационная деятельность. Информационные ресурсы
	2.2	Экономика информационной сферы
	2.3	Информационная этика и право, информационная безопасность
3		Средства ИКТ
	3.1	Архитектура компьютеров и компьютерных сетей
	3.1.1	Программная и аппаратная организация компьютеров и компьютерных систем. Виды программного обеспечения
	3.1.2	Операционные системы. Понятие о системном администрировании
	3.1.3	Безопасность, гигиена, эргономика, ресурсосбережение, технологические требования при эксплуатации компьютерного рабочего места
	3.2	Технологии создания и обработки текстовой информации
	3.2.1	Понятие о настольных издательских системах. Создание компьютерных публикаций
	3.2.2	Использование готовых и создание собственных шаблонов. Использование систем проверки орфографии и грамматики. Тезаурусы. Использование систем двуязычного перевода и электронных словарей
	3.2.3	Использование специализированных средств редактирования математических текстов и графического представления математических объектов
	3.2.4	Использование систем распознавания текстов
	3.3	Технология создания и обработки графической и мультимедийной информации
	3.3.1	Форматы графических и звуковых объектов
	3.3.2	Ввод и обработка графических объектов
	3.3.3	Ввод и обработка звуковых объектов
	3.4	Обработка числовой информации
	3.4.1	Математическая обработка статистических данных
	3.4.2	Использование динамических (электронных) таблиц для выполнения учебных заданий из различных предметных областей
	3.4.3	Использование инструментов решения статистических и расчетно-графических задач
	3.5	Технологии поиска и хранения информации
	3.5.1	Системы управления базами данных. Организация баз данных
	3.5.2	Использование инструментов поисковых систем (формирование запросов)

3.6	Телекоммуникационные технологии
3.6.1	Специальное программное обеспечение средств телекоммуникационных технологий
3.6.2	Инструменты создания информационных объектов для Интернета
3.7	Технологии управления, планирования и организации деятельности человека

Раздел 2. Перечень требований к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике и ИКТ

Перечень требований к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике и ИКТ, составлен с учетом сформулированных в образовательном стандарте целей изучения предмета, а также на основе раздела «Требования к уровню подготовки выпускников» Федерального компонента государственных стандартов среднего (полного) общего образования по информатике и ИКТ (базовый и профильный уровни).

В первом столбце даны коды требований, во втором столбце – требования к уровню подготовки выпускников, достижение которого проверяется на ЕГЭ.

Код требований	Проверяемые умения или способы действий
1	ЗНАТЬ/ПОНИМАТЬ/УМЕТЬ:
1.1	Моделировать объекты, системы и процессы
1.1.1	Проводить вычисления в электронных таблицах
1.1.2	Представлять и анализировать табличную информацию в виде графиков и диаграмм
1.1.3	Строить информационные модели объектов, систем и процессов в виде алгоритмов
1.1.4	Читать и отлаживать программы на языке программирования
1.1.5	Создавать программы на языке программирования по их описанию
1.1.6	Строить модели объектов, систем и процессов в виде таблицы истинности для логического высказывания
1.1.7	Вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний
1.2	Интерпретировать результаты моделирования
1.2.1	Использовать готовые модели, оценивать их соответствие реальному объекту и целям моделирования
1.2.2	Интерпретировать результаты, получаемые в ходе моделирования реальных процессов

1.3	Оценивать числовые параметры информационных объектов и процессов
1.3.1	Оценивать объем памяти, необходимый для хранения информации
1.3.2	Оценивать скорость передачи и обработки информации
2	ИСПОЛЬЗОВАТЬ ПРИОБРЕТЕННЫЕ ЗНАНИЯ И УМЕНИЯ В ПРАКТИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ И ПОВСЕДНЕВНОЙ ЖИЗНИ:
2.1	Осуществлять поиск и отбор информации
2.2	Создавать и использовать структуры хранения данных
2.3	Работать с распространенными автоматизированными информационными системами
2.4	Готовить и проводить выступления, участвовать в коллективном обсуждении, фиксировать его ход и результаты с использованием современных программных и аппаратных средств коммуникаций
2.5	Проводить статистическую обработку данных с помощью компьютера
2.6	Выполнять требования техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информатизации

Возможные алгоритмические задачи для подраздела 1.1 перечня требований к уровню подготовки выпускников, достижение которых проверяется на едином государственном экзамене по информатике и ИКТ.

- Нахождение минимума и максимума двух, трех, четырех данных чисел без использования массивов и циклов.
- Нахождение всех корней заданного квадратного уравнения.
- Запись натурального числа в позиционной системе с основанием, меньшим или равным 10. Обработка и преобразование такой записи числа.
- Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива).
- Использование цикла для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту и т.д.).
- Заполнение элементов одномерного и двумерного массивов по заданным правилам.
- Операции с элементами массива. Линейный поиск элемента. Вставка и удаление элементов в массиве. Перестановка элементов данного массива в обратном порядке. Суммирование элементов массива. Проверка соответствия элементов массива некоторому условию.
- Нахождение второго по величине (второго максимального или второго минимального) значения в данном массиве за однократный просмотр массива.

- Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.
- Операции с элементами массива, отобранных по некоторому условию (например, нахождение минимального четного элемента в массиве, нахождение количества и суммы всех четных элементов в массиве).
- Сортировка массива.
- Слияние двух упорядоченных массивов в один без использования сортировки.
- Обработка отдельных символов данной строки. Подсчет частоты появления символа в строке.
- Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

Спецификация
контрольных измерительных материалов
для проведения в 2015 году
единого государственного экзамена
по информатике и ИКТ

подготовлен Федеральным государственным бюджетным
научным учреждением

«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»

СПЕЦИФИКАЦИЯ
контрольных измерительных материалов
единого государственного экзамена 2015 года
по информатике и ИКТ

1. Назначение КИМ ЕГЭ

Единый государственный экзамен (далее – ЕГЭ) представляет собой форму объективной оценки качества подготовки лиц, освоивших образовательные программы среднего общего образования, с использованием заданий стандартизированной формы (контрольных измерительных материалов).

ЕГЭ проводится в соответствии с Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».

Контрольные измерительные материалы позволяют установить уровень освоения выпускниками Федерального компонента государственного стандарта среднего (полного) общего образования по информатике и ИКТ, базовый и профильный уровни.

Результаты единого государственного экзамена по информатике и ИКТ признаются образовательными организациями среднего профессионального образования и образовательными организациями высшего профессионального образования как результаты вступительных испытаний по информатике и ИКТ.

2. Документы, определяющие содержание КИМ ЕГЭ

Содержание экзаменационной работы определяет Федеральный компонент государственных стандартов среднего (полного) общего образования, базовый и профильный уровни (приказ Минобрнауки России от 05.03.2004 № 1089).

3. Подходы к отбору содержания, разработке структуры КИМ ЕГЭ

Содержание заданий разработано по основным темам курса информатики и ИКТ, объединенных в следующие тематические блоки: «Информация и ее кодирование», «Моделирование и компьютерный эксперимент», «Системы счисления», «Логика и алгоритмы», «Элементы теории алгоритмов», «Программирование», «Архитектура компьютеров и компьютерных сетей», «Обработка числовой информации», «Технологии поиска и хранения информации».

Содержанием экзаменационной работы охватывается основное содержание курса информатики и ИКТ, важнейшие его темы, наиболее значимый в них материал, однозначно трактуемый в большинстве преподаваемых в школе вариантов курса информатики и ИКТ.

Работа содержит как задания базового уровня сложности, проверяющие знания и умения, предусмотренные стандартом базового уровня, так

и задания повышенного и высокого уровней сложности, проверяющие знания и умения, предусмотренные стандартом профильного уровня. Количество заданий в варианте КИМ должно, с одной стороны, обеспечить всестороннюю проверку знаний и умений выпускников, приобретенных за весь период обучения по предмету, и, с другой стороны, соответствовать критериям сложности, устойчивости результатов, надежности измерения. С этой целью в КИМ используются задания двух типов: с кратким ответом и развернутым ответом. Структура экзаменационной работы обеспечивает оптимальный баланс заданий разных типов и разновидностей, трех уровней сложности, проверяющих знания и умения на трех различных уровнях: воспроизведения, применения в стандартной ситуации, применения в новой ситуации. Содержание экзаменационной работы отражает значительную часть содержания предмета. Все это обеспечивает валидность результатов тестирования и надежность измерения.

4. Структура КИМ ЕГЭ

Каждый вариант экзаменационной работы состоит из двух частей и включает в себя 27 заданий, различающихся формой и уровнем сложности.

Часть 1 содержит 23 задания с кратким ответом.

В экзаменационной работе предложены следующие разновидности заданий с кратким ответом:

- задания на выбор и запись одного или нескольких правильных ответов из предложенного перечня ответов;
- задания на вычисление определенной величины;
- задания на установление правильной последовательности, представленной в виде строки символов по определенному алгоритму.

Ответ на задания части 1 дается соответствующей записью в виде натурального числа или последовательности символов (букв и цифр), записанных без пробелов и других разделителей.

Часть 2 содержит 4 задания с развернутым ответом.

Часть 1 содержит 23 задания базового, повышенного и высокого уровней сложности. В этой части собраны задания с кратким ответом, подразумевающие самостоятельное формулирование и запись ответа в виде числа или последовательности символов. Задания проверяют материал всех тематических блоков. В части 1 12 заданий относятся к базовому уровню, 10 заданий к повышенному уровню сложности, 1 задание – к высокому уровню сложности.

Часть 2 содержит 4 задания, первое из которых повышенного уровня сложности, остальные 3 задания высокого уровня сложности. Задания этой части подразумевают запись развернутого ответа в произвольной форме.

Задания части 2 направлены на проверку сформированности важнейших умений записи и анализа алгоритмов, предусмотренных требованиями к обязательному уровню подготовки по информатике учащихся средних общеобразовательных учреждений. Эти умения проверяются на повышенном

и высоком уровнях сложности. Также на высоком уровне сложности проверяются умения по теме «Технология программирования».

Распределение заданий по частям экзаменационной работы представлено в таблице 1.

Таблица 1. Распределение заданий по частям экзаменационной работы

Часть работы	Количество заданий	Максимальный первичный балл	Процент максимального первичного балла за задания данной части от максимального первичного балла за всю работу, равного 35	Тип заданий
Часть 1	23	23	66	С кратким ответом
Часть 2	4	12	34	С развернутым ответом
Итого	27	35	100	

5. Распределение заданий КИМ по содержанию, видам умений и способам действий

Отбор содержания, подлежащего проверке в КИМ ЕГЭ 2015 г., осуществляется на основе Федерального компонента государственного стандарта среднего (полного) общего образования (базовый и профильный уровни). Распределение заданий по разделам курса информатики представлено в таблице 2.

Таблица 2. Распределение заданий по разделам курса информатики

№	Название раздела	Количество заданий	Максимальный первичный балл	Процент максимального первичного балла за задания данного вида от максимального первичного балла за всю работу, равного 35
1	Информация и ее кодирование	4	4	11
2	Моделирование и компьютерный эксперимент	2	2	6
3	Системы счисления	2	2	6
4	Логика и алгоритмы	6	8	23
5	Элементы теории алгоритмов	5	6	17
6	Программирование	4	9	25
7	Архитектура компьютеров и компьютерных сетей	1	1	3
8	Обработка числовой информации	1	1	3
9	Технологии поиска и хранения информации	2	2	6
	Итого	27	35	100

В КИМ ЕГЭ по информатике не включены задания, требующие простого воспроизведения знания терминов, понятий, величин, правил (такие задания слишком просты для выполнения). При выполнении любого из зада-

ний КИМ от экзаменуемого требуется решить тематическую задачу: либо прямо использовать известное правило, алгоритм, умение, либо выбрать из общего количества изученных понятий и алгоритмов наиболее подходящее и применить его в известной или новой ситуации.

Знание теоретического материала проверяется косвенно через понимание используемой терминологии, взаимосвязей основных понятий, размерностей единиц и т.д. при выполнении экзаменуемыми практических заданий по различным темам предмета. Таким образом, в КИМ по информатике и ИКТ проверяется освоение теоретического материала из разделов:

- единицы измерения информации;
- принципы кодирования;
- системы счисления;
- моделирование;
- понятие алгоритма, его свойств, способов записи;
- основные алгоритмические конструкции;
- основные понятия, используемые в информационных и коммуникационных технологиях.

Экзаменационная работа содержит одно задание, требующее прямо применить изученное правило, формулу, алгоритм. Это задание (4) отмечено как задание на воспроизведение знаний и умений.

Материал на проверку сформированности *умений применять свои знания в стандартной ситуации* входит обе части экзаменационной работы. Это следующие умения:

- анализировать однозначность двоичного кода;
- формировать для логической функции таблицу истинности и логическую схему;
- оперировать массивами данных;
- подсчитать информационный объем сообщения;
- искать кратчайший путь в графе, осуществлять обход графа;
- осуществлять перевод из одной системы счисления в другую;
- использовать стандартные алгоритмические конструкции при программировании;
- формально исполнять алгоритмы, записанные на естественных и алгоритмических языках, в том числе на языках программирования;
- определять мощность адресного пространства компьютерной сети по маске подсети в протоколе ТСП/Р;
- оценить результат работы известного программного обеспечения;
- формулировать запросы к базам данных и поисковым системам.

Материал на проверку сформированности *умений применять свои знания в новой ситуации* также входит в обе части экзаменационной работы. Это следующие сложные умения:

- анализировать обстановку исполнителя алгоритма;

- определять основание системы счисления по свойствам записи чисел;
- описывать свойства двоичной последовательности по алгоритму ее построения;
- осуществлять преобразования логических выражений;
- моделировать результаты поиска в сети Интернет;
- анализировать результат исполнения алгоритма;
- анализировать текст программы с точки зрения соответствия записанного алгоритма поставленной задаче и изменять его в соответствии с заданием;
- умение построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию;
- реализовывать сложный алгоритм с использованием современных систем программирования.

Распределение заданий по видам проверяемой деятельности представлено в таблице 3.

Таблица 3. Распределение заданий по проверяемым видам деятельности

Код	Виды деятельности	Количество заданий	Максимальный первичный балл	Процент максимального первичного балла за задания данного вида деятельности от максимального первичного балла за всю работу, равного 35
1	Воспроизведение представлений или знаний (при выполнении практических заданий)	1	1	3
2	Применение знаний и умений в стандартной ситуации	13	14	40
3	Применение знаний и умений в новой ситуации	13	20	57
Итого		27	35	100

Каждое задание экзаменационной работы характеризуется не только проверяемым содержанием, но и проверяемыми умениями. Кодификатор определяет две группы требований к уровню подготовки выпускников: с одной стороны, знать/понимать/уметь и, с другой стороны, использовать приобретенные знания и умения в практической деятельности и повседневной жизни.

При том что стандарт образования по информатике и ИКТ содержит достаточно много требований к использованию приобретенных знаний и умений в практической жизни, используемая стандартизированная бланковая технология единого государственного экзамена не позволяет проверить выполнение этих требований в полном объеме. В работе всего 3 таких задания, они расположены в части 1 работы. Их выполнение дает менее 10% первич-

ных баллов. Остальные 90% первичных баллов экзаменуемый может получить за счет реализации умений оперировать с теоретическим материалом предмета информатики. В таблице 4 характеризуется распределение заданий с точки зрения проверяемых умений в каждой части работы.

Таблица 4. Распределение заданий по видам умений и способам действий

Основные умения и способы действий	Количество заданий (процент максимального балла за выполнение заданий)		
	Вся работа	Часть 1 (задания с кратким ответом)	Часть 2 (задания с развернутым ответом)
1. Требования: «Знать/понимать/уметь»	24 (91%)	20 (57%)	4 (34%)
Моделирование объектов, систем и процессов	16 (68%)	12 (34%)	4 (34%)
Интерпретация результатов моделирования	4 (11,5%)	4 (11,5%)	0
Определение количественных параметров информационных процессов	4 (11,5%)	4 (11,5%)	0
2. Требования: «Использовать приобретенные знания и умения в практической деятельности и повседневной жизни»	3 (9%)	3 (9%)	0 (0%)
Осуществлять поиск и отбор информации	1 (3%)	1 (3%)	-
Создавать и использовать структуры хранения данных	1 (3%)	1 (3%)	-
Работать с распространенными автоматизированными информационными системами	1 (3%)	1 (3%)	-
Итого	27 (100%)	23 (66%)	4 (34%)

б. Распределение заданий КИМ по уровню сложности

Часть 1 экзаменационной работы содержит 12 заданий базового уровня сложности, 10 заданий повышенного уровня и 1 задание высокого уровня сложности.

Задания части 2 относятся к повышенному (1 задание) и высокому уровням. Предполагаемый процент выполнения заданий базового уровня – 60–90. Предполагаемый процент выполнения заданий повышенного уровня – 40–60. Предполагаемый процент выполнения заданий высокого уровня – менее 40.

Для оценки достижения базового уровня используются с кратким ответом. Достижение повышенного уровня подготовки проверяется с помощью заданий с кратким и развернутым ответами. Для проверки достижения высокого уровня подготовки в экзаменационной работе используются задания с кратким и развернутым ответами. Распределение заданий по уровням сложности представлено в таблице 5.

Таблица 5. Распределение заданий по уровням сложности

Уровень сложности заданий	Количество заданий	Максимальный первичный балл	Процент максимального первичного балла за задания данного вида деятельности от максимального первичного балла за всю работу, равного 35
Базовый	12	12	34
Повышенный	11	13	37
Высокий	4	10	29
Итого	27	35	100

Внутри каждой из двух частей работы задания расположены по принципу нарастающей сложности теста. Сначала идут задания базового уровня; затем – повышенного; затем – высокого. Задания одного уровня сложности расположены с учетом вида проверяемой деятельности и последовательности расположения тем в кодификаторе содержания.

7. Система оценивания выполнения отдельных заданий и экзаменационной работы в целом

Задания КИМ оцениваются разным количеством баллов в зависимости от их типа.

Выполнение каждого задания части 1 оценивается в 1 балл. Задание части 1 считается выполненным, если экзаменуемый дал ответ, соответствующий коду верного ответа. За выполнение каждого задания присваивается (в дихотомической системе оценивания) либо 0 баллов («задание не выполнено»), либо 1 балл («задание выполнено»). Ответы на задания части 1 автоматически обрабатываются после сканирования бланков ответов № 1.

Максимальное количество первичных баллов, которое можно получить за выполнение заданий части 1, – 23.

Выполнение заданий части 2 оценивается от 0 до 4 баллов. Ответы на задания части 2 проверяются и оцениваются экспертами. Максимальное количество баллов, которое можно получить за выполнение заданий части 2, – 12.

В соответствии с Порядком проведения государственной итоговой аттестации по образовательным программам среднего общего образования (приказ Минобрнауки России от 26.12.2013 № 1400 зарегистрирован Минюстом России 03.02.2014 № 31205)

«61. По результатам первой и второй проверок эксперты независимо друг от друга выставляют баллы за каждый ответ на задания экзаменационной работы ЕГЭ с развернутым ответом...

62. В случае существенного расхождения в баллах, выставленных двумя экспертами, назначается третья проверка. Существенное расхождение в баллах определено в критериях оценивания по соответствующему учебному предмету.

Эксперту, осуществляющему третью проверку, предоставляется информация о баллах, выставленных экспертами, ранее проверявшими экзаменационную работу».

Если расхождение составляет 2 и более балла за выполнение любого из заданий, то третий эксперт проверяет ответы только на те задания, которые вызвали столь существенное расхождение.

Максимальный первичный балл – 35. Баллы для поступления в вузы и вузы подсчитываются по 100-балльной шкале на основе анализа результатов выполнения всех заданий работы.

8. Продолжительность ЕГЭ по информатике и ИКТ

На выполнение экзаменационной работы отводится 3 часа 55 минут (235 минут). На выполнение заданий части 1 рекомендуется отводить 1,5 часа (90 минут). Остальное время рекомендуется отводить на выполнение заданий части 2.

Переход экзаменуемого от выполнения заданий части 1 к заданиям части 2 никак не фиксируется, последовательность выполнения заданий не регламентируется. Контроля времени выполнения отдельных заданий не ведется.

9. Изменения в КИМ 2015 года по сравнению с КИМ 2014 года

КИМ 2015 г. претерпел значительные изменения по сравнению с КИМ 2014 г. Изменена структура варианта КИМ: каждый вариант состоит из двух частей. Задания в варианте представлены в режиме сквозной нумерации без буквенных обозначений А, В, С.

Оптимизирована структура экзаменационной работы: сократилось общее количество заданий (с 32 до 27); соответственно, уменьшилось с 40 до 35 максимальное количество первичных баллов. Уменьшение количества заданий произведено за счет укрупнения тематики заданий, сведения близких по тематике и сложности заданий в одну позицию. Такой укрупненными стали позиция 3 (хранение информации в компьютере), 6 (формальное исполнение алгоритмов), 7 (технология вычислений и визуализации данных с помощью электронных таблиц) и 9 (скорость передачи звуковых и графических файлов). В связи с уменьшением количества частей, изменилась последовательность заданий в варианте. Часть 2 работы (задания с развернутым ответом) не изменилась, но относительный вес баллов, полученных за задания с развернутым ответом, увеличился за счет сокращения общего количества заданий в варианте.

Обобщенный план варианта КИМ ЕГЭ 2015 года по ИНФОРМАТИКЕ и ИКТ

Уровни сложности заданий: Б – базовый (примерный интервал выполнения задания – 60–90%); П – повышенный (40–60%); В – высокий (менее 40%).

№	Проверяемые элементы содержания	Коды проверяемых элементов содержания по кодификатору	Коды требований к уровню подготовки выпускников по кодификатору	Коды видов деятельности (п. 5 спецификации)	Уровень сложности задания	Макс. балл за выполнение задания	Примерное время выполнения задания (мин.)
1	Умение кодировать и декодировать информацию	1.1.2	1.2.2	2	Б	1	2
2	Умение строить таблицы истинности и логические схемы	1.5.1	1.1.6	2	Б	1	3
3	Знания о файловой системе организации данных или о технологии хранения, поиска и сортировки информации в базах данных	3.1.2/ 3.5.1	2.1/ 2.2	2	Б	1	3
4	Знания о системах счисления и двоичном представлении информации в памяти компьютера	1.4.2	1.3	1	Б	1	1
5	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	1.3.1	1.2.2	2	Б	1	3
6	Формальное исполнение алгоритма, записанного на естественном языке или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд	1.6.1/ 1.6.3	1.1.3	2	Б	1	4
7	Знание технологии обработки информации в электронных таблицах и методов визуализации данных с помощью диаграмм и графиков	3.4.1/ 3.4.3	1.1.1/ 1.1.2	2	Б	1	3
8	Знание основных конструкций языка программирования, понятия переменной, оператора присваивания	1.7.2	1.1.4	2	Б	1	3
9	Умение определять скорость передачи информации при заданной пропускной способности канала, объем памяти, необходимый для хранения звуковой и графической информации	1.1.4/ 3.3.1	1.3.1/ 1.3.2	3	Б	1	5
10	Знания о методах измерения количества информации	1.1.3	1.3.1	3	Б	1	4

11	Умение исполнить рекурсивный алгоритм	1.5.3	1.1.3	2	Б	1	5
12	Знание базовых принципов организации и функционирования компьютерных сетей, адресации в сети	3.1.1	2.3	2	Б	1	2
13	Умение подсчитывать информационный объем сообщения	1.1.3	1.3.1	2	П	1	3
14	Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд	1.6.2	1.2.2	3	П	1	6
15	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	1.3.1	1.2.1	2	П	1	3
16	Знание позиционных систем счисления	1.4.1	1.1.3	3	П	1	2
17	Умение осуществлять поиск информации в Интернете	3.5.2	2.1	3	П	1	2
18	Знание основных понятий и законов математической логики	1.5.1	1.1.7	3	П	1	3
19	Работа с массивами (заполнение, считывание, поиск, сортировка, массовые операции и др.)	1.5.2/ 1.5.6	1.1.4	2	П	1	5
20	Анализ алгоритма, содержащего вспомогательные алгоритмы, цикл и ветвление	1.6.1	1.1.4	3	П	1	5
21	Умение анализировать программу, использующую процедуры и функции	1.7.2	1.1.4	3	П	1	6
22	Умение анализировать результат исполнения алгоритма	1.6.2	1.1.3	3	П	1	7
23	Умение строить и преобразовывать логические выражения	1.5.1	1.1.7	3	В	1	10
24	Умение прочесть фрагмент программы на языке программирования и исправить допущенные ошибки	1.7.2	1.1.4	3	П	3	30
25	Умения написать короткую (10–15 строк) простую программу (например, обработки массива) на языке программирования или записать алгоритм на естественном языке	1.6.3	1.1.5	2	В	2	30
26	Умение построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию	1.5.2	1.1.3	3	В	3	30
27	Умения создавать собственные программы (30–50 строк) для решения задач средней сложности	1.7.3	1.1.5	3	В	4	55
<p>Всего заданий – 27; из них по типу заданий: с кратким ответом – 23; с развернутым ответом – 4; по уровню сложности: Б – 12, П – 11, В – 4. Максимальный первичный балл за работу – 35. Общее время выполнения работы – 235 мин.</p>							