

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

Демонстрационный вариант
контрольных измерительных материалов единого
государственного экзамена 2016 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
2016 года по ИНФОРМАТИКЕ и ИКТ**

При ознакомлении с демонстрационным вариантом контрольных измерительных материалов ЕГЭ 2016 г. следует иметь в виду, что задания, включённые в него, не отражают всех вопросов содержания, которые будут проверяться с помощью вариантов КИМ в 2016 г. Полный перечень вопросов, которые могут контролироваться на едином государственном экзамене 2016 г., приведён в кодификаторе элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения единого государственного экзамена 2016 г. по информатике и ИКТ.

Назначение демонстрационного варианта заключается в том, чтобы дать возможность любому участнику ЕГЭ и широкой общественности составить представление о структуре будущих КИМ, количестве заданий, об их форме 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.

Демонстрационный вариант
контрольных измерительных материалов
для проведения в 2016 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 1–23 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: **23.**

23

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, или капиллярной, или перьевой ручек.

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

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

Желаем успеха!

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

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

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

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

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

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

Часть 1

Ответами к заданиям 1–23 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.

1 Сколько единиц в двоичной записи шестнадцатеричного числа 12F0₁₆?

Ответ: _____.

2 Логическая функция F задаётся выражением $(\neg z) \wedge x \vee x/y$. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z .

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

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала – буква, соответствующая 1-му столбцу; затем – буква, соответствующая 2-му столбцу; затем – буква, соответствующая 3-му столбцу). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Пусть задано выражение $x \rightarrow y$, зависящее от двух переменных x и y , и таблица истинности:

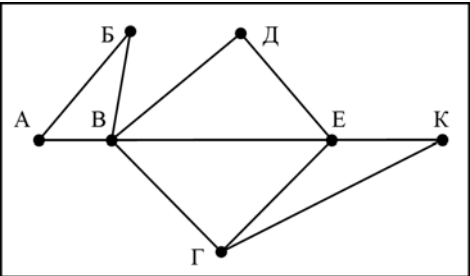
Перем. 1	Перем. 2	Функция
???	???	F
0	0	1
0	1	1
1	0	0
0	1	1

Тогда 1-му столбцу соответствует переменная y , а 2-му столбцу соответствует переменная x . В ответе нужно написать: yx .

Ответ: _____.

3 На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1		45		10			
п2	45			40		55	
п3					15	60	
п4	10	40				20	35
п5			15			55	
п6		55	60	20	55		45
п7				35		45	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта В в пункт Е. В ответе запишите целое число – так, как оно указано в таблице.

Ответ: _____.

4 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите, сколько прямых потомков (т.е. детей и внуков) Павленко А.К. упомянуты в таблице 1.

Таблица 1		
ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симонян А.А.	Ж
2511	Сокол В.А.	Ж
3193	Биба С.А.	Ж
...

Таблица 2	
ID Родителя	ID Ребёнка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511
2529	3193
...	...

Ответ: _____.

ИЛИ

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

В каталоге находится 6 файлов:

maveric.map
maveric.mp3
taverna.mp4
revolver.mp4
vera.mp3
zveri.mp3

Ниже представлено восемь масок. Сколько из них таких, которым соответствуют ровно четыре файла из данного каталога?

ver.mp*	*?ver?*.*mp?	?*ver*.mp?*	*v*r*?.m?p*
???*???.mp*	???*???.m*	*a*.*a*	*a*.*p*

Ответ: _____.

5

По каналу связи передаются сообщения, содержащие только четыре буквы: П, О, С, Т; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв Т, О, П используются такие кодовые слова: Т: 111, О: 0, П: 100.

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

Ответ: _____.

6

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

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 125. В ответе это число запишите в десятичной системе счисления.

Ответ: _____.

ИЛИ

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 2,

2. умножь на 5.

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

Например, программа 2121 – это программа

умножь на 5,

прибавь 2,

умножь на 5,

прибавь 2,

которая преобразует число 1 в число 37.

Запишите порядок команд в программе, которая преобразует **число 2 в число 24** и содержит не более четырёх команд. Указывайте лишь номера команд.

Ответ: _____.

7 Дан фрагмент электронной таблицы. Из ячейки E4 в ячейку D3 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Каким стало числовое значение формулы в ячейке D3?

	A	B	C	D	E
1	40	4	400	70	7
2	30	3	300	60	6
3	20	2	200		5
4	10	1	100	40	= \$B2 * C\$3

Примечание: знак \$ обозначает абсолютную адресацию.

Ответ: _____.

ИЛИ

Дан фрагмент электронной таблицы.

	A	B	C
1		6	10
2	=(A1-3)/(B1-1)	=(A1-3)/(C1-5)	=C1/(A1-3)

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 следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre>DIM S, N AS INTEGER S = 0 N = 0 WHILE S < 111 S = S + 8 N = N + 2 WEND PRINT N</pre>	<pre>s = 0 n = 0 while s < 111: s = s + 8 n = n + 2 print(n)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач цел n, s n := 0 s := 0 нц пока s < 111 s := s + 8 n := n + 2 кц вывод n кон</pre>	<pre>var s, n: integer; begin s := 0; n := 0; while s < 111 do begin s := s + 8; n := n + 2; end; writeln(n) end.</pre>
Си	
<pre>#include<stdio.h> int main() { int s = 0, n = 0; while (s < 111) { s = s + 8; n = n + 2; } printf("%d\n", n); return 0; }</pre>	

Ответ: _____.

- 9** Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 64×64 пикселей при условии, что в изображении могут использоваться 256 различных цветов? В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

ИЛИ

Музыкальный фрагмент был записан в формате моно, оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла – 24 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 4 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

- 10** Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 5-буквенные слова, в которых есть только буквы П, И, Р, причём буква П появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Ответ: _____.

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

Бейсик	Python
<pre> DECLARE SUB F(n) DECLARE SUB G(n) SUB F(n) IF n > 0 THEN G(n - 1) END SUB SUB G(n) PRINT "*" IF n > 1 THEN F(n - 3) END SUB </pre>	<pre> def F(n): if n > 0: G(n - 1) def G(n): print("*") if n > 1: F(n - 3) </pre>
Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 0 то G(n - 1) все кон алг G(цел n) нач вывод "*" если n > 1 то F(n - 3) все кон </pre>	<pre> procedure F(n: integer); forward; procedure G(n: integer); forward; procedure F(n: integer); begin if n > 0 then G(n - 1); end; procedure G(n: integer); begin writeln('*'); if n > 1 then F(n - 3); end; </pre>
Си	
<pre> void F(int n); void G(int n); void F(int n){ if (n > 0) G(n - 1); } void G(int n){ printf("*"); if (n > 1) F(n - 3); } </pre>	

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?

Ответ: _____.

12 В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда – нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: _____.

13 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: A, B, C, D, E, F, G, H, K, L, M, N. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 20 пользователях потребовалось 400 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Ответ: _____.

14 Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) заменить (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды **заменить** (111, 27)

преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки v , то выполнение команды **заменить** (v, w) не меняет эту строку.

Б) нашлось (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА *условие*

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда1*

ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

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

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 68 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА **нашлось** (222) ИЛИ **нашлось** (888)

ЕСЛИ **нашлось** (222)

ТО **заменить** (222, 8)

ИНАЧЕ **заменить** (888, 2)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

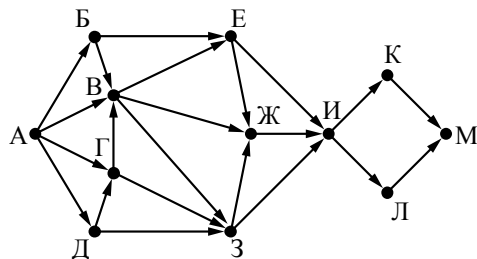
КОНЕЦ

Ответ: _____.

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

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

Сколько существует различных путей из города А в город М?



Ответ: _____.

- 16** Значение арифметического выражения: $9^8 + 3^5 - 9$ – записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

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 тысячах)
<i>Гомер & Илиада</i>	200
<i>Гомер & (Одиссея Илиада)</i>	470
<i>Гомер & Одиссея</i>	355

Какое количество страниц (в тысячах) будет найдено по запросу *Гомер & Одиссея & Илиада*?

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

Ответ: _____.

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

Для какого наименьшего неотрицательного целого числа A формула

$$x \& 25 \neq 0 \rightarrow (x \& 17 = 0 \rightarrow x \& A \neq 0)$$

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

Ответ: _____.

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

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

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

Ответ: _____.

- 20** Ниже на пяти языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает число M . Известно, что $x > 100$. Укажите **наименьшее** такое (т.е. большее 100) число x , при вводе которого алгоритм печатает 26.

Бейсик	Python
<pre> DIM X, L, M AS INTEGER INPUT X L = X M = 65 IF L MOD 2 = 0 THEN M = 52 ENDIF WHILE L <> M IF L > M THEN L = L - M ELSE M = M - L ENDIF WEND PRINT M </pre>	<pre> x = int(input()) L = x M = 65 if L % 2 == 0: M = 52 while L != M: if L > M: L = L - M else: M = M - L print(M) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел x, L, M ввод x L := x M := 65 если mod(L, 2) = 0 то M := 52 все нц пока L <> M если L > M то L := L - M иначе M := M - L все кц вывод M кон </pre>	<pre> var x, L, M: integer; begin readln(x); L := x; M := 65; if L mod 2 = 0 then M := 52; while L <> M do if L > M then L := L - M else M := M - L; writeln(M); end. </pre>

Си
<pre>#include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L = x; M = 65; if (L % 2 == 0) M = 52; while (L != M){ if(L > M) L = L - M; else M = M - L; } printf("%d", M); }</pre>

Ответ: _____.

21

Напишите в ответе наименьшее значение входной переменной k , при котором программа выдаёт тот же ответ, что и при входном значении $k=10$. Для Вашего удобства программа приведена на пяти языках программирования.

Бейсик	Python
<pre>DIM K, I AS LONG INPUT K I = 1 WHILE F(I) < G(K) I = I + 1 WEND PRINT I FUNCTION F(N) F = N * N * N END FUNCTION FUNCTION G(N) G = 2*N + 3 END FUNCTION</pre>	<pre>def f(n): return n*n*n def g(n): return 2*n+3 k = int(input()) i = 1 while f(i) < g(k): i+=1 print (i)</pre>

Алгоритмический язык	Паскаль
<pre>алг нач цел i, k ввод k i := 1 нц пока f(i) < g(k) i := i + 1 кц вывод i кон алг цел f(цел n) нач знач := n * n * n кон алг цел g(цел n) нач знач := 2*n + 3 кон</pre>	<pre>var k, i : longint; function f(n: longint): longint; begin f := n * n * n; end; function g(n: longint): longint; begin g := 2*n + 3; end; begin readln(k); i := 1; while f(i) < g(k) do i := i+1; writeln(i) end.</pre>
Си	
<pre>#include<stdio.h> long f(long n) { return n * n * n; } long g(long n) { return 2*n + 3; } int main() { long k, i; scanf("%ld", &k); i = 1; while(f(i)<g(k)) i++; printf("%ld", i); return 0; }</pre>	

Ответ: _____.

- 22** Исполнитель Май15 преобразует число на экране.
У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1

2. Умножить на 2

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.
Программа для исполнителя Май15 – это последовательность команд.

Сколько существует программ, для которых при исходном числе 2 результатом является число 29 и при этом траектория вычислений содержит число 14 и не содержит числа 25?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **121** при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Ответ: _____.

- 23** Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_9, y_1, y_2, \dots, y_9$, которые удовлетворяют всем перечисленным ниже условиям?

$$(\neg(x_1 \equiv y_1)) \equiv (x_2 \equiv y_2)$$

$$(\neg(x_2 \equiv y_2)) \equiv (x_3 \equiv y_3)$$

...

$$(\neg(x_8 \equiv y_8)) \equiv (x_9 \equiv y_9)$$

В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, \dots, x_9, y_1, y_2, \dots, y_9$, 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 положительное целое число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, DIGIT, SUM AS LONG INPUT N SUM = 0 WHILE N > 0 DIGIT = N MOD 10 IF DIGIT < 7 THEN SUM = SUM + 1 END IF N = N \ 10 WEND PRINT DIGIT </pre>	<pre> N = int(input()) sum = 0 while N > 0: digit = N % 10 if digit < 7: sum = sum + 1 N = N // 10 print(digit) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел N, digit, sum ввод N sum := 0 нц пока N > 0 digit := mod(N,10) если digit < 7 то sum := sum + 1 все N := div(N,10) кц вывод digit кон </pre>	<pre> var N, digit, sum: longint; begin readln(N); sum := 0; while N > 0 do begin digit := N mod 10; if digit < 7 then sum := sum + 1; N := N div 10; end; writeln(digit) end. </pre>

Си

```
#include <stdio.h>
int main()
{
    int N, digit, sum;
    scanf("%d", &N);
    sum = 0;
    while (N > 0)
    {
        digit = N % 10;
        if (digit < 7)
            sum = sum + 1;
        N = N / 10;
    }
    printf("%d",digit);
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 456.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный 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, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

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

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

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

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

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) **один** камень или увеличить количество камней в куче в **два раза**. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 73. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 73 камня или больше.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (6, 34), (7, 33), (9, 32) выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций (6, 33), (8, 32) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (6, 32), (7, 32), (8, 31) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (7, 31) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

27 В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь. Необходимо вычислить «бета-значение» серии показаний прибора – минимальное **чётное** произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным -1 .

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

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

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

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

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ А.

Максимальная оценка за выполнение задания А – 2 балла.

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

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

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

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ Б.

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

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

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

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

Пример входных данных:

11
12
45
5
3
17
23
21
20
19
18
17

Программа должна вывести одно число – описанное в условии произведение либо -1 , если получить такое произведение не удаётся.

Пример выходных данных для приведённого выше примера входных данных:

54

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	6
2	zyx
3	20
4	7 3
5	101
6	31 1211
7	8 8
8	28
9	4 128
10	80
11	3
12	192
13	12
14	28
15	56
16	3
17	85
18	8
19	2
20	130
21	3
22	13
23	1024

Часть 2

Критерии оценивания заданий с развёрнутым ответом

24

На обработку поступает положительное целое число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, DIGIT, SUM AS LONG INPUT N SUM = 0 WHILE N > 0 DIGIT = N MOD 10 IF DIGIT < 7 THEN SUM = SUM + 1 END IF N = N \ 10 WEND PRINT DIGIT </pre>	<pre> N = int(input()) sum = 0 while N > 0: digit = N % 10 if digit < 7: sum = sum + 1 N = N // 10 print(sum) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел N, digit, sum ввод N sum := 0 нц пока N > 0 digit := mod(N,10) если digit < 7 то sum := sum + 1 все N := div(N,10) кц вывод digit кон </pre>	<pre> var N, digit, sum: longint; begin readln(N); sum := 0; while N > 0 do begin digit := N mod 10; if digit < 7 then sum := sum + 1; N := N div 10; end; writeln(digit) end. </pre>

Сн

```
#include <stdio.h>
int main()
{
    int N, digit, sum;
    scanf("%d", &N);
    sum = 0;
    while (N > 0)
    {
        digit = N % 10;
        if (digit < 7)
            sum = sum + 1;
        N = N / 10;
    }
    printf("%d", digit);
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 456.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный 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 любом из четырёх других языков.

1. Программа выведет число 4.

2. Пример числа, при вводе которого программа выдаёт верный ответ: 835.

Замечание для проверяющего. Программа работает неправильно из-за неверной выводимой на экран переменной и неверного увеличения суммы.

Соответственно, программа будет работать верно, если в числе старшая цифра (крайняя левая) равна сумме цифр, меньших 7.

3. В программе есть две ошибки.

Первая ошибка. Неверное увеличение суммы.

Строка с ошибкой:

`sum := sum + 1;`

Верное исправление:

`sum := sum + digit;`

Вторая ошибка. Неверный вывод ответа на экран.

Строка с ошибкой:

`writeln(digit)`

Верное исправление:

`writeln(sum)`

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 конкретном входном числе; 2) указать пример входного числа, при котором программа выдаёт верный ответ; 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
Максимальный балл	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, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

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

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

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

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

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

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

На языке Си	
<pre>k = 0; for (i = 0; i < N-1; i++) if (a[i]%3 == 0 a[i+1]%3 == 0) k++; printf("%d", k);</pre>	
На языке Python	
<pre>k = 0 for i in range(0, n - 1): if (a[i] % 3 == 0 or a[i + 1] % 3 == 0): k += 1 print(k)</pre>	
На естественном языке	
<p>Записываем в переменную <i>K</i> начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления текущего и следующего элемента массива на 3. Если первый или второй из полученных остатков равен 0, увеличиваем переменную <i>K</i> на единицу. После завершения цикла выводим значение переменной <i>K</i></p>	
Указания по оцениванию	Баллы
<p><i>Общие указания.</i></p> <p>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</p> <p>2. Эффективность алгоритма не имеет значения и не оценивается.</p> <p>3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования; при этом количество переменных и их идентификаторы должны соответствовать условию задачи</p>	

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

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) **один** камень или увеличить количество камней в куче в **два раза**. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 73. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 73 камня или больше.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (6, 34), (7, 33), (9, 32) выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций (6, 33), (8, 32) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (6, 32), (7, 32), (8, 31) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (7, 31) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Задание 1. В начальных позициях (6, 33), (8, 32) выигрышная стратегия есть у Вани. При начальной позиции (6, 33) после первого хода Пети может получиться одна из следующих четырёх позиций: (7, 33), (12, 33), (6, 34), (6, 66). Каждая из этих позиций содержит менее 73 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 73 камней, удвоив количество камней во второй куче. Для позиции (8, 32) после первого хода Пети может получиться одна из следующих четырёх позиций: (9, 32), (16, 32), (8, 33), (8, 64). Каждая из этих позиций содержит менее 73 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 73 камней, удвоив количество камней во второй куче. Таким образом, Ваня при любом ходе Пети выигрывает своим первым ходом.</p> <p>Задание 2. В начальных позициях (6, 32), (7, 32) и (8, 31) выигрышная стратегия есть у Пети. При начальной позиции (6, 32) он должен первым ходом получить позицию (6, 33), из начальных позиций (7, 32) и (8, 31). Петя после первого хода должен получить позицию (8, 32). Позиции (6, 33) и (8, 32) рассмотрены при разборе задания 1. В этих позициях выигрышная стратегия есть у игрока, который будет ходить вторым (теперь это Петя). Эта стратегия описана при разборе задания 1. Таким образом, Петя при любой игре Вани выигрывает своим вторым ходом.</p> <p>Задание 3. В начальной позиции (7, 31) выигрышная стратегия есть у Вани. После первого хода Пети может возникнуть одна из четырёх позиций: (8, 31), (7, 32), (14, 31) и (7, 62). В позициях (14, 31) и (7, 62) Ваня может выиграть одним ходом, удвоив количество камней во второй куче. Позиции (8, 31) и (7, 32) были рассмотрены при разборе задания 2. В этих позициях у игрока, который должен сделать ход (теперь это Ваня), есть выигрышная стратегия. Эта стратегия описана при разборе задания 2. Таким образом, в зависимости от игры Пети Ваня выигрывает на первом или втором ходу.</p> <p><i>Примечание для эксперта.</i> Последняя фраза в приведённом решении избыточна. Не будет ошибкой, если экзаменуемый просто напишет, например, «При выбранной стратегии партия длится не более двух ходов».</p> <p>В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключение позиции (в них выигрывает Ваня) выделены жирным шрифтом.</p>

Исходное положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(7, 31) Всего: 38	(7, 31+1) = (7, 32) Всего: 39	(7+1, 32) = (8, 32) Всего: 40	(8+1, 32) = (9, 32) Всего: 41	(9, 32*2) = (9, 64) Всего: 73
			(8, 32+1) = (8, 33) Всего: 41	(8, 33*2) = (8, 66) Всего: 74
			(8*2, 32) = (16, 32) Всего: 48	(16, 32*2) = (16, 64) Всего: 80
			(8, 32*2) = (8, 64) Всего: 72	(8, 64*2) = (8, 128) Всего: 136
	(7+1, 31) = (8, 31) Всего: 39	(8, 31+1) = (8, 32) Всего: 40	(8+1, 32) = (9, 32) Всего: 41	(9, 32*2) = (9, 64) Всего: 73
			(8, 32+1) = (8, 33) Всего: 41	(8, 33*2) = (8, 66) Всего: 74
			(8*2, 32) = (16, 32) Всего: 48	(16, 32*2) = (16, 64) Всего: 80
			(8, 32*2) = (8, 64) Всего: 72	(8, 64*2) = (8, 128) Всего: 136
	(7*2, 31) = (14, 31) Всего: 45	(14, 31*2) = (14, 62) Всего: 76		
	(7, 31*2) = (7, 62) Всего: 69	(7, 62*2) = (7, 124) Всего: 131		

Примечание для эксперта. Дерево всех партий может быть также изображено в виде ориентированного графа – так, как показано на рисунке, или другим способом. Например, вершины дерева, соответствующие одной и той

же позиции, на рисунке могут быть «склеены». Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

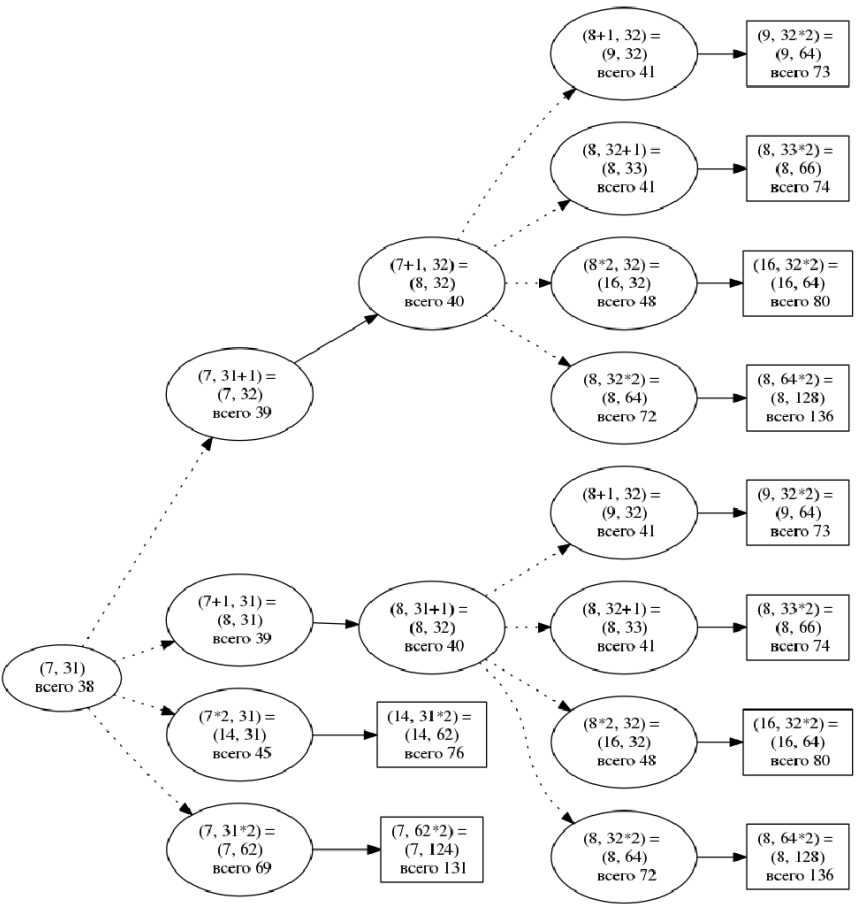


Рис. 1. Дерево всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны пунктирными стрелками, ходы Вани показаны сплошными стрелками. Заключительные позиции обозначены прямоугольником.

Примечание для эксперта. В некоторых позициях у Вани есть и другой способ выигрыша: например, в позиции (8, 64) можно добавить один камень в любую кучу. То, что это не указано, не является ошибкой. Экзаменуемый не должен указывать все возможные выигрышные стратегии

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить три задания. 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. Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены все три задания. 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 хотя бы одно из следующих условий. <ul style="list-style-type: none">– Выполнено задание 3.– Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий. <ul style="list-style-type: none">– Выполнено задание 1.– Выполнено задание 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
Максимальный балл	3

27

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное **чётное** произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным -1 .

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

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

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

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

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ А.

Максимальная оценка за выполнение задания А – 2 балла.

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

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

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

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ Б.

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

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

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

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

Пример входных данных:

```
11
12
45
5
3
17
23
21
20
19
18
17
```

Программа должна вывести одно число – описанное в условии произведение либо -1 , если получить такое произведение не удаётся.

Пример выходных данных для приведённого выше примера входных данных:

```
54
```

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

Задание Б (решение для задания А приведено ниже, см. программу 4). Чтобы произведение было чётным, хотя бы один сомножитель должен быть чётным, поэтому при поиске подходящих произведений чётные показания прибора можно рассматривать в паре с любыми другими, а нечётные – только с чётными.

Для каждого показания с номером k , начиная с $k = 7$, рассмотрим все допустимые по условиям задачи пары, в которых данное показание получено вторым. Минимальное произведение из всех этих пар будет получено, если первым в паре будет взято минимальное подходящее показание среди всех, полученных от начала приёма и до показания с номером $k - 6$. Если очередное показание чётное, минимальное среди предыдущих может быть любым, если нечётное – только чётным.

Для получения эффективного по времени решения нужно по мере ввода данных помнить абсолютное минимальное и минимальное чётное показание на каждый момент времени, каждое вновь полученное показание умножать на соответствующий ему минимум, имевшийся на 6 элементов ранее, и выбрать минимальное из всех таких произведений.

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

Ниже приводится пример такой программы, написанной на алгоритмическом языке.

Программа 1. Пример правильной программы на алгоритмическом языке.
Программа эффективна по времени и по памяти

```

алг
нач
  цел s = 6          | требуемое расстояние между показаниями
  цел аmax = 1001    | больше максимально возможного показания
  цел N
  ввод N
  цел а
  целтаб мини[0:s-1] | очередное показание прибора
  целтаб миничет[0:s-1] | текущие минимумы последних s элементов
  цел i
  | вводим первые s показаний, фиксируем минимумы
  цел ма; ма := аmax | минимальное показание
  цел мчет; мчет := аmax | минимальное чётное показание
  нц для i от 1 до s
    ввод а
    ма := imin(ма, а)
    если mod(a,2) = 0 то мчет := imin(мчет,а) все
    мини[mod(i, s)] := ма
    миничет[mod(i, s)] := мчет
  кц
  цел мп = аmax*аmax | минимальное значение произведения
  цел п
  нц для i от s+1 до N
    ввод а
    если mod(a,2)=0
      то п := а * мини[mod(i, s)]
      иначе если мчет < аmax
        то п := а * миничет[mod(i, s)]
        иначе п := аmax*аmax;
      все
    все
    мп := imin(мп, п)
    ма := imin(ма, а)
    если mod(a,2) = 0 то мчет := imin(мчет,а) все
    мини[mod(i, s)] := ма
    миничет[mod(i, s)] := мчет
  кц
  если мп = аmax*аmax то мп:=-1 все
  вывод мп
кон

```

Возможны и другие реализации. Например, вместо циклического заполнения массива можно каждый раз сдвигать его элементы. В приведённом ниже примере хранятся и сдвигаются не минимумы, а исходные значения. Это требует чуть меньше памяти (достаточно одного массива вместо двух), но по времени решение со сдвигами менее эффективно, чем с циклическим заполнением. Однако время работы остаётся пропорциональным N , поэтому максимальная оценка за такое решение тоже составляет 4 балла.

Программа 2. Пример правильной программы на языке Паскаль.**Программа использует сдвиги, но эффективна по времени и по памяти**

```

const s = 6; {требуемое расстояние между показаниями}
      amax = 1001; {больше максимально возможного показания}
var
  N: integer;
  a: array[1..s] of integer; {хранение s показаний прибора}
  a_: integer; {ввод очередного показания}
  ma: integer; {минимальное число без s последних}
  me: integer; {минимальное чётное число без s последних}
  mp: integer; {минимальное значение произведения}
  p: integer;
  i, j: integer;
begin
  readln(N);
  {Ввод первых s чисел}
  for i:=1 to s do readln(a[i]);
  {Ввод остальных значений, поиск минимального произведения}
  ma := amax; me := amax;
  mp := amax*amax;
  for i := s + 1 to N do begin
    readln(a_);
    if a[1] < ma then ma := a[1];
    if (a[1] mod 2 = 0) and (a[1] < me) then me := a[1];
    if a_ mod 2 = 0 then p := a_ * ma
    else if me < amax then p := a_ * me
    else p := amax* amax;
    if (p < mp) then mp := p;
    {сдвигаем элементы вспомогательного массива влево}
    for j := 1 to s - 1 do
      a[j] := a[j + 1];
    a[s] := a_
  end;
  if mp = amax*amax then mp:=-1;
  writeln(mp)
end.

```

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

Программа 3. Пример правильной программы на языке Паскаль.**Программа эффективна по времени, но неэффективна по памяти**

```

const s = 6; {требуемое расстояние между показаниями}
      amax = 1001; {больше максимально возможного показания}
var
  N, p, i: integer;
  a: array[1..10000] of integer; {все показания прибора}
  ma: integer; {минимальное число без s последних}
  me: integer; {минимальное чётное число без s последних}
  mp: integer; {минимальное значение произведения}
begin
  readln(N);
  {Ввод всех показаний прибора}
  for i:=1 to N do readln(a[i]);
  ma := amax;
  me := amax;
  mp := amax*amax;
  for i := s + 1 to N do
    begin
      if a[i-s] < ma then ma := a[i-s];
      if (a[i-s] mod 2 = 0) and (a[i-s] < me) then
        me := a[i-s];
      if a[i] mod 2 = 0 then p := a[i] * ma
      else if me < amax then p := a[i] * me
      else p := amax * amax;
      if (p < mp) then mp := p
    end;
    if mp = amax*amax then mp := -1;
    writeln(mp)
  end.

```

Возможно также переборное решение, в котором находятся произведения всех возможных пар и из них выбирается минимальное. Ниже (см. программу 4) приведён пример подобного решения. Это (и аналогичные ему) решение неэффективно ни по времени, ни по памяти. Оно является решением задания А, но не является решением задания Б. Оценка за такое решение – 2 балла.

Программа 4. Пример правильной программы на языке Паскаль.
Программа неэффективна ни по времени, ни по памяти

```
const s = 6; {требуемое расстояние между показаниями}
var
  N: integer;
  a: array[1..10000] of integer; {все показания прибора}
  mp: integer; {минимальное значение произведения}
  i, j: integer;
begin
  readln(N);
  {Ввод значений прибора}
  for i:=1 to N do
    readln(a[i]);
  mp := 1000 * 1000 + 1;
  for i := 1 to N-s do begin
    for j := i+s to N do begin
      if (a[i]*a[j] mod 2 = 0) and (a[i]*a[j] < mp)
        then mp := a[i]*a[j]
      end;
    end;
  if mp = 1000 * 1000 + 1 then mp := -1;
  writeln(mp)
end.
```

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

Пояснения для проверяющих.

1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то согласно приведённым ниже критериям его оценка будет такой же, как если бы это решение было представлено в качестве решения задания А.

2. Два задания (и, соответственно, возможность для экзаменуемого представить две программы) дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое даёт ему право получить 2 балла, а затем приступить к поиску более эффективного решения.

3. Приведённые в п. 2.1–2.5 правила имеют целью избежать снижения оценки из-за того, что ученик перепутал обозначения заданий

Критерии оценивания задания А

Программа решает поставленную задачу для любых соответствующих условию входных данных. Например, допускается переборное решение, аналогичное приведённой выше программе 4.

Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла).

Допускается до двух содержательных ошибок, описанных в критериях оценивания задания Б на 3 балла

Не выполнены условия, позволяющие поставить 2 балла.

Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях

Не выполнены критерии, позволяющие поставить 1 или 2 балла

Максимальный балл для задания А

Критерии оценивания задания Б

Программа правильно работает для любых соответствующих условию входных данных и при этом эффективна как по времени, так и по памяти, т.е. не используются массивы и другие структуры данных, размер которых зависит от количества входных элементов, а время работы пропорционально этому количеству. Возможно использование массивов и динамических структур данных (например, контейнеры STL в программе на языке C++) при условии, что в них в каждый момент времени хранится фиксированное количество элементов, требующих для хранения меньше 1кб (минимально необходимое количество – шесть; допускается решение с запасом).

Программа может содержать не более трёх синтаксических ошибок следующих видов:

- 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, недопустимая для соответствующего типа данных.

К синтаксическим ошибкам приравнивается использование неверного типа данных (например, использование целого типа вместо вещественного для представления данных при вводе и обработке).

Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку

Не выполнены условия, позволяющие поставить 4 балла.

Программа правильно работает для любых соответствующих условию входных данных, время работы пропорционально количеству входных элементов. Размер используемой памяти не имеет значения и может зависеть от объёма входных данных. В частности, допускается использование одного или нескольких массивов размера N (как в приведённой выше программе 3).

Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла. Кроме того, допускается наличие не более одной содержательной ошибки из числа следующих:

- неверная инициализация при поиске минимального значения;
- неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при $6 < N < 12$;

<ul style="list-style-type: none"> – 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 операция "<" вместо "<=", "or" вместо "and" и т.п.; – не учитывается, что заданные показания могут начинаться с одного или нескольких чётных чисел; – не учитывается, что для данного набора показаний может не быть ни одного удовлетворяющего условиям произведения 	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла.</p> <p>Программа работает в целом верно, эффективно или нет. Например, допускается переборное решение, аналогичное приведённой выше программе 4.</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>	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 быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях</p>	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл для задания Б</i>	4
<i>Итоговый максимальный балл</i>	4

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 развёрнутым 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.

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 раз- дела	Код контро- лируемого 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 2016 году
единого государственного экзамена
по информатике и ИКТ

подготовлена Федеральным государственным бюджетным
научным учреждением

«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»

СПЕЦИФИКАЦИЯ
контрольных измерительных материалов
единого государственного экзамена 2016 года
по информатике и ИКТ

1. Назначение КИМ ЕГЭ

Единый государственный экзамен (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 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 экзамена 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 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 или 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. 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 2016 г., осуществляется на основе Федерального компонента государственного стандарта среднего (полного) общего образования (базовый и профильный уровни). Распределение заданий по разделам курса информатики и ИКТ представлено в таблице 2.

Таблица 2. Распределение заданий по разделам курса информатики и ИКТ

№	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 и ИКТ не включены задания, требующие простого воспроизведения знания терминов, понятий, величин, правил (такие задания слишком просты для выполнения). При выполнении любого из заданий 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 (1) отмечено как задание на воспроизведение знаний и умений.

Материал на проверку сформированности *умений применять свои знания в стандартной ситуации* входит в 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 TCP/IP;
- 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	Вид 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% первичных баллов. Остальные 90% первичных баллов экзаменуемый может получить за счет реализации умений оперировать с теоретическим материалом предмета информатики и ИКТ. 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%)

6. 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. 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.

8. Дополнительные материалы и оборудование

Дополнительные материалы и оборудование не используются.

9. 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 24–27, то третий эксперт проверяет ответы только на те задания, которые вызвали столь существенное расхождение.

Максимальный первичный балл – 35.

Баллы для поступления в вузы подсчитываются по 100-балльной шкале на основе анализа результатов выполнения всех заданий работы.

10. Изменения в КИМ 2016 года по сравнению с КИМ 2015 года

Модель КИМ 2016 г. по сравнению с КИМ 2015 г. изменилась незначительно. Была изменена последовательность предъявления заданий 1–5.

Количество заданий и максимальный первичный балл остались без изменений.

Обобщенный план варианта КИМ ЕГЭ 2016 года по ИНФОРМАТИКЕ и ИКТ

Уровни сложности заданий: Б – базовый; П – повышенный; В – высокий.

№	Проверяемые элементы содержания	Коды проверяемых элементов со- держания по коди- фика- тору	Коды требо- ваний к уровню подго- товки выпу- сков по коди- фика- тору	Коды ви- дов- дея- тель- ности (п. 5 спе- ци- фи- кации)	Уро- вень слож- ности зада- ния	Макс. балл за выпол- нение зада- ния	При- мер- ное вре- мя вы- пол- нения зада- ния (мин.)
Часть 1							
1	Знания о системах счисления и двоичном представлении информации в памяти компьютера	1.4.2	1.3	1	Б	1	1
2	Умения строить таблицы истинности и логические схемы	1.5.1	1.1.6	2	Б	1	3
3	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
4	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
5	Умение кодировать и декодировать информацию	1.1.2	1.2.2	2	Б	1	2
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 сети Интернет	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 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

Часть 2

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 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

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**.