

**Единый государственный экзамен
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 развернутым 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. Ответы запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.

КИМ

Ответ: 23.

1	2	3																							
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Бланк

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

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

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

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

Желаем успеха!

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связей (операций):

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

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

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

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

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, чье 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 Дано четыре выражения: 1011111_2 , 61_{16} , 137_8 , $3C_{16}+44_8$

Какое из них имеет наибольшее значение? В ответе запишите это значение в десятичной системе счисления, основание системы счисления писать не нужно.

Ответ: _____.

2 При заполнении таблицы истинности логической функции $\neg(z \rightarrow y) \vee w \vee \neg x$ Андрей успел заполнить лишь фрагмент из семи различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z.

???	???	???	???	$\neg(z \rightarrow y) \vee w \vee \neg x$
0	0	1	0	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z.

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

		$\neg x \vee y$
0	1	

то первому столбцу соответствовала бы переменная y, а второму столбцу – переменная x. В ответе следовало бы написать ux.

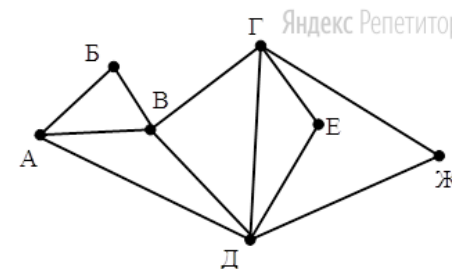
Ответ: _____.

3 Схему дорог некоторого района изобразили в виде графа. После этого в таблицу занесли информацию о длинах тех же самых дорог (в километрах). Отсутствие числа в таблице означает, что соответствующей дороги не существует.

	1	2	3	4	5	6	7
1				15	17	12	33
2					18		28
3					9		13
4	15				23	10	
5	17	18	9	23			25
6	12			10			
7	33	28	13		25		

Так как таблицу и схему рисовали независимо друг друга разные исполнители, то цифровые обозначения населенных пунктов в таблице никак не связаны с буквенными обозначениями на графе. Однако известно, что каждому населенному пункту на схеме соответствует один номер в таблице.

Определите, каково кратчайшее расстояние из пункта Г в пункт Д. В ответе запишите это расстояние без указания наименования.



Ответ: _____.



4 Ниже представлены фрагменты двух таблиц из базы данных о жителях некоторого поселка. Каждая строка таблицы 2 содержит информацию о ребенке и об одном из его родителей.

Таблица 1

ID	Фамилия И.О.	Пол	Год рождения
110110	Афанасов Б.В.	м	1981
111111	Афанасов М.Б.	м	2005
112112	Афанасова А.Б.	ж	2007
113113	Афанасова М.И.	ж	1982
114114	Волошин И.И.	м	2000
115115	Волошин С.И.	м	1998
116116	Волошина М.С.	ж	1975
117117	Горьковец А.М.	ж	1995
118118	Горьковец Е.А.	м	1993
119119	Горьковец О.Е.	ж	2018
120120	Горьковец С.Е.	ж	2016
121121	Горьковец Я.Е.	м	2014
145145	Кучко А.К.	ж	2019
146146	Кучко К.И.	м	1997
147147	Кучко С.Б.	ж	2001

Таблица 2

ID родителя	ID ребенка
110	111
110	112
110	147
113	111
113	112
113	147
116	114
116	115
116	117
117	119
117	120
117	121
118	119
118	120
118	121
146	145
147	145

На основании приведенных данных определите количество женщин, имеющих не менее двух детей в возрасте до 16 лет (включительно) по состоянию на 2019 год. При вычислении ответа учитывайте только информацию из приведенных фрагментов таблиц.

Ответ: _____.

5 Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, Д, Е, К решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовали кодовое слово 1; для буквы Б – кодовое слово 01.

Какова наименьшая возможная сумма длин кодовых слов для букв В, Г, Д, Е, К?

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

Ответ: _____.

6 Автомат получает на вход два двузначных шестнадцатеричных числа. По этим числам строится новое шестнадцатеричное число по следующим правилам.

- 1) Вычисляются два шестнадцатеричных числа – сумма старших разрядов исходных чисел и сумма младших разрядов.
- 2) Полученные два шестнадцатеричных числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходные числа: 66, 43. Поразрядные суммы: А, 9. Результат: 9А. Определите, какое количество чисел из списка ниже может быть результатом работы автомата.

810 123 5E 213 1C A6 F1F 1B1E

Ответ: _____.





7 Дан фрагмент электронной таблицы. Из ячейки B2 в ячейку D3 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились.

	A	B	C	D	E
1	1	10	100	1000	5
2	2	= \$C\$2+B3+5	200	2000	6
3	3	30	300		7
4	4	40	400	4000	8

Каким стало числовое значение формулы в ячейке D3? *Примечание.* Знак \$ обозначает абсолютную адресацию.

Ответ: _____.

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

C++	Python	Паскаль
<pre>#include <iostream> using namespace std; int main() { int a = 7; int b = 10 + a * 2; while (b < 100) { a = b + 12; b = a + 14; } cout << b << endl; return 0; }</pre>	<pre>a = 7 b = 10 + a * 2 while b < 100: a = b + 12 b = a + 14 print(b)</pre>	<pre>var a, b: integer; begin a := 7; b := 10 + a * 2; while b < 100 do begin a := b + 12; b := a + 14; end; writeln(b); end.</pre>
Бейсик	Алгоритмический язык	
<pre>DIM A AS INTEGER DIM B AS INTEGER A = 7 B = 10 + a * 2 WHILE B < 100 A = B + 12 B = A + 14 WEND</pre>	<pre>алг нач цел a, b a := 7 b := 10 + a * 2 нц пока b < 100 a := b + 12 b := a + 14 кц</pre>	

PRINT B	кц вывод b кон	
---------	----------------------	--

Ответ: _____.

9 Светлана сделала аудиозапись доклада своей одноклассницы и хочет переслать полученный файл по Wi-Fi-каналу связи. Параметры аудиозаписи: режим записи — стерео (двухканальная), частота дискретизации — 16 кГц, число уровней квантования сигнала — 65536, длительность записи — 8 минут. При записи данных в файл сжатие данных не производилось. Известно, что максимально возможная скорость передачи данных по имеющемуся в наличии Светланы каналу связи составляет 10 Мбит в секунду.

Определите, какое минимальное количество секунд может потребоваться Светлане для передачи файла.
В ответе укажите целое число секунд.

Примечание. 1 Мбит = 2²⁰ бит.

Ответ: _____.

10 Все пятибуквенные слова, составленные из букв Ф, А, Й, Л, записаны в алфавитном порядке и пронумерованы, начиная с 1. (Словом считается последовательность букв, необязательно осмысленная.) Начало списка выглядит так:

- 1) ААААА
- 2) ААААЙ
- 3) ААААЛ
- 4) ААААФ
- 5) АААЙА
- 6) АААЙЙ

...
Под каким номером в списке идёт первое слово, которое начинается с букв ЛФ?

Ответ: _____.

11 Ниже на четырех языках программирования записан рекурсивный алгоритм F.

C++	Python	Паскаль
<pre>void F(int n){ if (n > 0){ std::cout <<n << endl; F(n - 4); F(n / 4); } }</pre>	<pre>def F(n): if n > 0: print(n) F(n - 4) F(n // 4)</pre>	<pre>procedure F(n: integer); begin if n > 0 then begin writeln(n); F(n - 4); F(n div 4) end end;</pre>
Бейсик		
<pre>SUB F(n as INTEGER) IF n > 0 THEN PRINT n F(n - 4) F(n \ 4) END IF END SUB</pre>		

В программе произведен вызов F(10). Чему будет равна сумма чисел, выведенных на экран?
В качестве ответа запишите одно целое число.

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 231.191.37.84 адрес сети равен 231.191.37.80. Определите максимально возможное число различных адресов компьютеров в данной сети, если два адреса (адрес сети и широковещательный) не используют?

Ответ запишите в виде десятичного числа.

Ответ: _____.

13 При регистрации в компьютерной системе предприятия каждому пользователю выдаётся электронный пропуск, на котором записаны личный код сотрудника, информация о подразделении и некоторая дополнительная информация.
Личный код состоит из некоторого количества знаков, одинакового для каждого сотрудника.
Каждый знак может быть символом из 26-символьного набора прописных латинских букв или десятичной цифрой. Личный код записывается на пропуске в виде целого числа байт, при этом каждый символ личного кода кодируется одинаковым минимально возможным количеством бит.
Информация о подразделении – это целое число в диапазоне от 1 до 73. Информация о подразделении также записывается в виде целого числа байт.
Для дополнительной информации в каждом пропуске отведено 16 байт.
В базе данных для хранения сведений об электронном пропуске каждого сотрудника отведено одинаковое целое число байт.
Для хранения сведений об электронных пропусках 150 пользователей потребовалось 3600 байт.
Какое максимальное количество символов может быть в личном коде сотрудника? В ответе запишите целое число.

Ответ: _____.

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

A) 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 vv 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
КОНЕЦ ЕСЛИ

выполняется команда1 (если 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 100100 идущих подряд букв А?

НАЧАЛО
ПОКА нашлось (ВВ) ИЛИ нашлось (ААА)
ЕСЛИ нашлось (ААА)
ТО заменить (ААА, В)
ИНАЧЕ заменить (ВВ, А)
КОНЕЦ ЕСЛИ

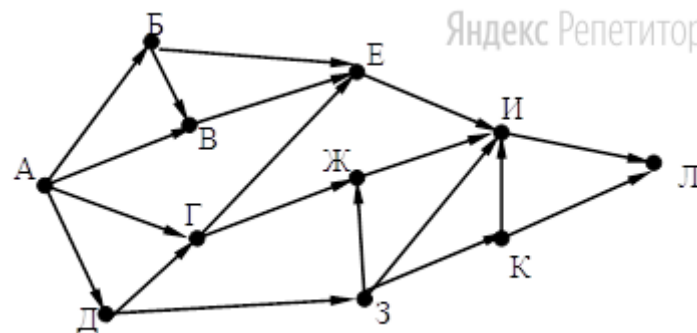
КОНЕЦ ПОКА
КОНЕЦ

В ответе запишите полученную строку.

Ответ: _____.

15

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



Сколько существует различных путей из города А в город Л, проходящих через город И?

Ответ: _____.

16

Значение выражения $9^{2019} + 3^{2019} \cdot 3^{19} + 2019$ 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	Количество страниц (тыс.)
Роза	200
Шиповник	140
Календула	96
Роза Календула Шиповник	250
Календула & Шиповник	0
Роза & Шиповник	100

Сколько страниц (в тысячах) будет найдено по запросу *Роза & Календула* ?

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

Ответ: _____.

18 Определите наименьшее неотрицательное целое число А, такое, что формула

$$(x \cdot y + A > 10) \vee (x < 25) \wedge (A < y)$$

тождественно истинна (т.е. принимает значение 1) при любых неотрицательных целых значениях переменных x и y.

Ответ: _____.

19 В программе используется одномерный целочисленный массив АА с индексами от 0 до 9.

Значения элементов равны 1,2,3,5,7,3,10,17,6,4 соответственно т.е. $A[0]=1, A[1]=2$ и т.д.

Определите значение переменной k после выполнения следующего фрагмента программы (записанного ниже на четырёх языках программирования).

C++	Python	Паскаль
<pre>k=0; for (j=1; j<= 9; j++) if (A[j] > A[0]) {t = A[j]; A[j] = A[0]; A[0] = t*2; k = k + 1; }</pre>	<pre>k = 0 for j in range(1,10): if A[j] > A[0]: t = A[j] A[j] = A[0] A[0] = t * 2 k = k + 1</pre>	<pre>k:=0; for j := 1 to 9 do if A[j] > A[0] then begin t := A[j]; A[j] := A[0]; A[0] := t*2; k := k + 1; end;</pre>
Бейсик		
<pre>k = 0 FOR j = 1 TO 9 IF A(j) > A(0) THEN t = A(j) A(j) = A(0) A(0) = t * 2 k = k + 1 END IF NEXT j</pre>		

Ответ: _____.

20 Ниже на четырёх языках программирования записан алгоритм.

C++	Python	Паскаль
<pre>#include <iostream> using namespace std; int main() {int x, L, M; cin >> x; L = 1; M = 0; while (x > 0) { M = M + 1; if(x % 2 != 0) { L = L * (x % 8); } x = x / 8; }</pre>	<pre>x = int(input()) L = 1 M = 0 while x > 0: M = M + 1 if x % 2 != 0: L = L * (x % 8) x = x // 8 print(L) print(M)</pre>	<pre>var x, L, M: integer; begin readln(x); L := 1; M := 0; while x > 0 do begin M := M + 1; if x mod 2 <> 0 then L := L * (x mod 8); x := x div 8 end; writeln(L);</pre>

<pre>cout << L << endl << M << endl; return 0; }</pre>		<pre>writeln(M) end.</pre>
Бейсик		
<pre>DIM X, L, M AS INTEGER INPUT X L = 1 M = 0 WHILE X > 0 M = M + 1 IF X MOD 2 <> 0 THEN L = L * (X MOD 8) END IF X = X \ 8 WEND PRINT L PRINT M</pre>		

Получив на вход натуральное десятичное число x, этот алгоритм печатает два числа: L и M.

Укажите **наибольшее** число x, при вводе которого алгоритм печатает сначала 35, а потом 3.

Ответ: _____.

21 Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма, представленного на разных языках программирования.

C++	Python	Паскаль
<pre>#include <iostream> using namespace std; long F(long x) {return 2*(x-7)*(x-7)+7*x+7; } long G(long x)</pre>	<pre>def F(x): return 2*(x-7)*(x-7)+7*x+7 def G(x): return 3*x*x - 107 n = 20</pre>	<pre>var n, a, t:longint; function F(x: longint): longint; begin F:= 2*(x-7)*(x-7)+7*x+7; end;</pre>

<pre>{return 3*x*x - 107; } int main() {long a = 0, n; n = 20; for (int t = 1; t <= 20; t++) {if (F(n) >= G(t)) a++; } cout << a << endl; return 0; }</pre>	<pre>a = 0 for t in range(1,21): if (F(n) >= G(t)): a += 1 print (a)</pre>	<pre>function G(x: longint): longint; begin G:= 3*x*x - 107; end; begin n:=20; a:=0; for t:= 1 to 20 do begin if (F(n) >= G(t)) then a:=a+1; end; writeln(a); end.</pre>
Бейсик		
<pre>DIM A AS LONG DIM T AS LONG DIM N AS LONG FUNCTION F (x AS INTEGER) AS LONG F = 2*(x-7)*(x-7)+7*x+7 END FUNCTION FUNCTION G (x AS INTEGER) AS LONG G = 3*x*x - 107 END FUNCTION A = 0: N = 20 FOR T = 1 TO 20 IF F(N) >= G(T) THEN A = A + 1 END IF NEXT T PRINT A</pre>		

Ответ: _____.



ТРЕНИРОВОЧНЫЙ КИМ №191111



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

- 1) Прибавить 3
- 2) Умножить на 2

Первая из них увеличивает число на экране на 3, вторая умножает его на 2. Программа для Вычислителя – это последовательность команд.

Сколько существует таких программ, которые преобразуют исходное число **2** в число **46** и при этом траектория вычислений программы не содержит число **20**?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 1212 при исходном числе 5 траектория будет состоять из чисел 8, 16, 19, 28.

Ответ: _____.

23 Сколько существует различных наборов логических переменных x_1, \dots, x_{14} , которые удовлетворяют всем перечисленным ниже условиям?

$$\neg(x_1 \equiv x_2) \vee \neg(x_3 \equiv x_4) \vee \neg(x_5 \equiv x_6) = 1$$

$$\neg(x_5 \equiv x_6) \vee \neg(x_7 \equiv x_8) \vee \neg(x_9 \equiv x_{10}) = 1$$

$$\neg(x_9 \equiv x_{10}) \vee \neg(x_{11} \equiv x_{12}) \vee \neg(x_{13} \equiv x_{14}) = 1$$

$$\neg((x_1 \wedge x_5) \equiv (x_9 \wedge x_{13})) = 1$$

В ответе не нужно перечислять все различные наборы значений переменных, при которых выполнено данное равенство. В качестве ответа 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 Дано натуральное число N , не превышающее 10000. Необходимо вывести разницу между максимальной и минимальной цифрой в этом числе. Программист написал программу, которая иногда работает не правильно. Ниже эта программа для Вашего удобства приведена на разных языках программирования.

Python	Паскаль
<pre>n = int(input()) min = 0 max = 0 while n > 9: d = n % 10 if d < min: min = d if d > max: max = d n = n // 10 print(max - min)</pre>	<pre>var n, min, max, d: integer; begin read(n); min:=0; max:=0; while n>9 do begin d:= n mod 10; if d < min then min:=d; if d > max then max:=d; n := n div 10; end; writeln (max - min); end.</pre>
C++	Бейсик
<pre>#include <iostream> using namespace std; int main() { int n, d, min, max; cin >> n; min = 0; max = 0; while (n > 9) {d = n % 10; if (d < min) min = d; if (d > max) max = d; n = n / 10; } cout << max - min << endl; return 0; }</pre>	<pre>DIM n AS INTEGER DIM d AS INTEGER DIM min AS INTEGER DIM max AS INTEGER INPUT N min = 0: max = 0 WHILE n > 9 d = n MOD 10 IF d < min THEN min = d IF d > max THEN max = d n = n / 10 WEND PRINT max - min</pre>



Последовательно выполните следующее.

- 1) Напишите, что выведет эта программа при вводе числа 723723.
- 2) Приведите пример такого натурального числа, при вводе которого приведённая программа, несмотря на ошибки, выдаёт верный ответ.
- 3) Найдите допущенные программистом ошибки и исправьте их.

Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Для каждой ошибки:

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

Известно, что в тексте программы можно исправить ровно две строки так, чтобы она стала работать правильно.

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

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

25 Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от -10 000 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который должен находиться на месте многоточия и находить максимальное четное значение среди элементов массива, а затем заменять каждый четный элемент массива числом, равным найденному максимуму.

Гарантируется, что хотя бы одно четное значение среди элементов массива есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводить с новой строчки.

Например, для массива из шести элементов:

4 -115 7 195 25 -106

программа должна вывести:

44
-115
7
195
25
4

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

Python	Паскаль
<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(n): a.append(int(input())) ...</pre>	<pre>const N = 30; var a: array [1..N] of longint; i, j, k: longint; begin for i := 1 to N do readln(a[i]); ... end.</pre>
Бейсик	C++
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG, J AS LONG, K AS LONG FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>#include <iostream> using namespace std; const int N = 30; int main() { long a[N]; long i, j, k; for (i = 0; i < N; i++) cin >> a[i]; ... return 0; }</pre>



26

Два игрока, Петя и Вова, играют в следующую игру. Дан набор слов, составленных из букв русского алфавита, при этом ни одно из заданных слов не является началом другого. Слова в этой игре – это просто цепочки букв, они не обязаны быть осмысленными словами русского языка. Игра состоит в том, что игроки составляют слово из набора, приписывая по очереди буквы к концу составляемого слова, т.е. справа. При этом каждое промежуточное слово должно быть началом одного из заданных слов. Выигрывает тот, кто получит одно из заданных слов целиком. Первый ход делает Петя, т.е. Петя пишет первую букву составляемого слова.

Пример. Заданный набор слов: {АНТАРКТИДА, АНТРАЦИТ, АБАРА, АБАЖУР, БББ, БАОБАБ, БАРА}. Первым ходом Петя пишет Б (он мог написать Б или А). Вова в ответ дописывает А и получает БА (он мог ещё получить ББ). Вторым ходом Петя получает БАРА и выигрывает.

В заданиях используются следующие понятия. *Стратегия игрока* – это правило, указывающее игроку ход, который он должен сделать. *Описать стратегию* игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Стратегия игрока называется *выигрышной*, если игрок выигрывает в любой партии, разыгранной в соответствии с этой стратегией, как бы ни играл противник.

Множество всех партий, которые могут получиться при данной стратегии, представляется в виде дерева, это дерево называется *деревом всех партий для заданной стратегии*. В узлах дерева – позиции игры; на рёбрах – ходы, которые переводят одну позицию в другую; корень дерева – начальная позиция игры. Дерево всех партий для данной стратегии можно описать с помощью рисунка или таблицы.

Задание 1

- А. Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ГДЕЖЗИКЛ, КЛМНБВГ}. Опишите эту стратегию.
- Б. Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ДВАДВА...ДВА, ПОРАПОРА...ПОРА} (в первом слове ДВА повторено 3333 раза, т.е. его длина 9999 букв; во втором слове ПОРА повторено 2222 раза, т.е. его длина 8888 букв). Опишите эту стратегию.

Задание 2

В задании 1А поменяйте местами две буквы в более коротком слове так, чтобы теперь выигрышная стратегия была у другого игрока. Напишите полученный набор слов; опишите выигрышную стратегию.

Задание 3

Рассмотрим набор слов {СТОЛ, СТОЛЕНИЕ, СПОРТ, КОЛЕСО, КОЛБА, КАК}. У кого из игроков есть выигрышная стратегия для этого набора? Приведите в виде рисунка или таблицы дерево всех партий, возможных при этой стратегии.

27

На вход программы поступает последовательность из NV целых положительных чисел. Известно, что все числа в последовательности различны. Необходимо составить программу, которая подсчитывает количество пар различных элементов последовательности, произведение которых делится на 51. Элементы пары не обязательно должны стоять в последовательности рядом, порядок элементов в паре не важен.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 10000$).

В каждой из последующих NV строк записано одно целое положительное число, не превышающее 10 000.

В качестве результата программа должна напечатать одно число: количество пар, в которых произведение элементов кратно 51.

Пример входных данных:

5
1
2
3
17
51

Пример выходных данных для приведённого выше примера входных данных:

5
Пояснение. Из пяти заданных чисел можно составить 10 попарных произведений:

1·2 1·3 1·17 1·51 2·3 2·17 2·51 3·17

Из них на 51 делятся 5 произведений

1·51 2·51 3·17 3·51 17·51

Требуется написать эффективную по времени и по памяти программу для решения описанной задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел N в k раз время работы программы увеличивается не более чем в k раз.



Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 килобайта и не увеличивается с ростом N .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать одну программу или две программы решения задачи (например, одна из программ может быть менее эффективна). Если Вы сдадите две программы, то каждая из них будет оцениваться независимо от другой, итоговой станет большая из двух оценок.

Перед текстом программы обязательно кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

Alex Zharkov <https://vk.com/zharkovalan>

Dmitrii Marinin https://vk.com/dmitrii_marinin

О проекте «Пробный ЕГЭ каждую неделю»

Данный ким составлен командой всероссийского волонтерского проекта «ЕГЭ 100 баллов» <https://vk.com/ege100ballov> и безвозмездно распространяется для любых некоммерческих образовательных целей.

Нашли ошибку в варианте?

Напишите нам, пожалуйста, и мы обязательно её исправим!

Для замечаний и пожеланий: https://vk.com/topic-10175642_39951777

(также доступны другие варианты для скачивания)

Корректоры варианты

Виктория Луковская <https://vk.com/luivilla>

Патимат Захарова <https://vk.com/patimatzakharova>

Ольга Моисеева https://vk.com/moiseeva_olga

Любовь Выходец https://vk.com/lav_school2

Светлана Черноглаз <https://vk.com/id8675984>

Николай Путько https://vk.com/omni_k



Система оценивания экзаменационной работы по информатике и ИКТ

Часть 1

№ задания	Ответ
1	97
2	zxwy
3	22
4	2
5	22
6	5
7	4205
8	102
9	24
10	705
11	21
12	14
13	9
14	ВАА
15	10
16	2004
17	86
18	11
19	3
20	501
21	7
22	12
23	4272

Часть 2

24

Дано натуральное число N, не превышающее 10000. Необходимо вывести разницу между максимальной и минимальной цифрой в этом числе. Программист написал программу, которая иногда работает не правильно. Ниже эта программа для Вашего удобства приведена на разных языках программирования.

Python	Паскаль
<pre>n = int(input()) min = 0 max = 0 while n > 9: d = n % 10 if d < min: min = d if d > max: max = d n = n // 10 print(max - min)</pre>	<pre>var n, min, max, d: integer; begin read(n); min:=0; max:=0; while n>9 do begin d:= n mod 10; if d < min then min:=d; if d > max then max:=d; n := n div 10; end; writeln (max - min); end.</pre>
C++	Бейсик
<pre>#include <iostream> using namespace std; int main() { int n, d, min, max; cin >> n; min = 0; max = 0; while (n > 9) {d = n % 10; if (d < min) min = d; if (d > max) max = d; n = n / 10; } cout << max - min << endl; return 0; }</pre>	<pre>DIM n AS INTEGER DIM d AS INTEGER DIM min AS INTEGER DIM max AS INTEGER INPUT N min = 0: max = 0 WHILE n > 9 d = n MOD 10 IF d < min THEN min = d IF d > max THEN max = d n = n / 10 WEND PRINT max - min</pre>

Последовательно выполните следующее.

- 1) Напишите, что выведет эта программа при вводе числа 723723.



- 2) Приведите пример такого натурального числа, при вводе которого приведённая программа, несмотря на ошибки, выдаёт верный ответ.
- 3) Найдите допущенные программистом ошибки и исправьте их.

Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Для каждой ошибки:

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

Известно, что в тексте программы можно исправить ровно две строки так, чтобы она стала работать правильно.

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

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

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 языков программирования. <ol style="list-style-type: none"> 1. При вводе числа 723 программа выведет число 3. 2. Программа выдаёт правильный ответ, например, для числа 109. Замечание для проверяющего. Программа будет выдавать правильный ответ для чисел, у которых в старшем разряде стоит не максимальная цифра и в записи числа есть цифра «0». 3. В программе есть две ошибки. Первая ошибка: неверная инициализация переменной min. Строка с ошибкой: min:=0; Вариант верного исправления для программ на Паскале: min:=10; Другие варианты: min:=9; min:=11 и т.д. Вторая ошибка: Неверное условие продолжения цикла. Строка с ошибкой: while n>9 do

Указания по оцениванию		Баллы
Верное исправление: while n>0 do (В программах на других языках – исправления аналогичны)		
В задаче требуется выполнить четыре действия. <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. 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. Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: <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	
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. <ol style="list-style-type: none"> 1. Выполнены три из четырех необходимых действий. Ни одна верная строка не названа ошибочной. 2. Выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки. 	2	
Не выполнены условия, позволяющие поставить 2 или 3 балла.	1	
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0	
<i>Максимальный балл</i>	3	

25

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от -10 000 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который должен находиться на месте многоточия и находить максимальное четное значение среди



ТРЕНИРОВОЧНЫЙ КИМ №191111



элементов массива, а затем заменять каждый четный элемент массива числом, равным найденному максимуму.

Гарантируется, что хотя бы одно четное значение среди элементов массива есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводить с новой строки.

Например, для массива из шести элементов:

4 -115 7 195 25 -106

программа должна вывести:

44
-115
7
195
25
4

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

Python	Паскаль
<pre># опускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(n): a.append(int(input())) ...</pre>	<pre>const N = 30; var a: array [1..N] of longint; i, j, k: longint; begin for i := 1 to N do readln(a[i]); ... end.</pre>
Бейсик	C++
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG, J AS LONG, K AS LONG FOR I = 1 TO N INPUT A(I)</pre>	<pre>#include <iostream> using namespace std; const int N = 30; int main() { long a[N]; long i, j, k; for (i = 0; i < N; i++) cin >> a[i];</pre>

NEXT I ... END	... return 0; }
----------------------	-----------------------

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>C++</p> <pre>k = -10001; for (i = 0; i < N; i++) if ((a[i]%2 == 0) && (a[i]>k)) k=a[i]; for (i = 0; i < N; i++) { if (a[i]%2 == 0) a[i] = k; cout << a[i]<<endl; }</pre>	<p>Паскаль</p> <pre>k:= -10001; {поиск четного максимума} for i := 1 to N do if (a[i] mod 2 = 0) and (a[i] > k) then k := a[i]; {замена четных элементов и вывод всех элементов на экран} for i := 1 to N do begin if (a[i] mod 2 = 0) then a[i] := k; writeln(a[i]); end;</pre>
Указания по оцениванию	Баллы
<p><i>Общие указания</i></p> <ol style="list-style-type: none"> В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы. Эффективность алгоритма не имеет значения и не оценивается. 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 Допускается формат вывода массива, отличный от указанного, например, в строчку. 	



Предложен правильный алгоритм, выдающий в качестве результата верное значение	2
Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих: 1) в цикле происходит выход за границу массива; 2) не инициализируется или неверно инициализируется переменная для хранения максимума; 3) неверно осуществляется поиск максимума; 4) проверяется делимость на 2 не элемента массива, а его индекса; 5) неверно осуществляется поиск максимума; 6) неверно составлено логическое условие (например, используется <code>or</code> вместо <code>and</code>) 7) исходный массив не изменяется; 8) отсутствует вывод ответа или ответ выводится не полностью; 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 или меняется неверно	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

26

Два игрока, Петя и Вова, играют в следующую игру. Дан набор слов, составленных из букв русского алфавита, при этом ни одно из заданных слов не является началом другого. Слова в этой игре – это просто цепочки букв, они не обязаны быть осмысленными словами русского языка. Игра состоит в том, что игроки составляют слово из набора, приписывая по очереди буквы к концу составляемого слова, т.е. справа. При этом каждое промежуточное слово должно быть началом одного из заданных слов. Выигрывает тот, кто получит одно из заданных слов целиком. Первый ход делает Петя, т.е. Петя пишет первую букву составляемого слова.

Пример. Заданный набор слов: {АНТАРКТИДА, АНТРАЦИТ, АБАРА, АБАЖУР, БББ, БАОБАБ, БАР}. Первым ходом Петя пишет Б (он мог написать Б или А). Вова в ответ дописывает А и получает БА (он мог ещё получить ББ). Вторым ходом Петя получает БАР и выигрывает.

В заданиях используются следующие понятия. *Стратегия игрока* – это правило, указывающее игроку ход, который он должен сделать. *Описать стратегию* игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Стратегия игрока называется *выигрышной*, если игрок выигрывает в любой партии, разыгранной в соответствии с этой стратегией, как бы ни играл противник.

Множество всех партий, которые могут получиться при данной стратегии, представляется в виде дерева, это дерево называется *деревом всех партий для заданной стратегии*. В узлах дерева – позиции игры; на рёбрах – ходы, которые переводят одну позицию в другую; корень дерева – начальная позиция игры. Дерево всех партий для данной стратегии можно описать с помощью рисунка или таблицы.

Задание 1

- В. Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ГДЕЖЗИКЛ, КЛМНБВГ}. Опишите эту стратегию.
- Г. Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ДВАДВА...ДВА, ПОРАПОРА...ПОРА} (в первом слове ДВА повторено 3333 раза, т.е. его длина 9999 букв; во втором слове ПОРА повторено 2222 раза, т.е. его длина 8888 букв). Опишите эту стратегию.

Задание 2

В задании 1А поменяйте местами две буквы в более коротком слове так, чтобы теперь выигрышная стратегия была у другого игрока. Напишите полученный набор слов; опишите выигрышную стратегию.

Задание 3

Рассмотрим набор слов {СТОЛ, СТОЛЕНИЕ, СПОРТ, КОЛЕСО, КОЛБА, КАК}. У кого из игроков есть выигрышная стратегия для этого набора? Приведите в виде рисунка или таблицы дерево всех партий, возможных при этой стратегии.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
Общая идея решения Если в наборе только одно слово, то все ходы игроков единственно возможные. Если длина слова нечётная, то выигрывает Петя; если длина чётная – Вова.





Аналогично, пусть для некоторого слова существует единственная возможность продолжить его до слова из набора. Если длина продолжения нечётная, то выигрывает тот игрок, который ходит; если длина чётная, выигрывает его противник.

Задание 1

А. Выигрышная стратегия есть у Пети. Первым ходом он пишет К. В конце игры у игроков должно получиться слово КЛМНБВГ. Длина слова равна 77 (нечетная), поэтому последнюю букву напишет Петя.

Б. Выигрышная стратегия есть у Пети. Первым ходом он пишет Д. После этого у игроков должно получиться слово ДВАДВА...ДВА. В этом слове 9999 букв. Петя напишет 11-ю, 33-ю, 55-ю и т.д. буквы. Поэтому 9999-ю букву напишет Петя, и, следовательно, выигрывает Петя.

Задание 2

Нужно в слове КЛМНБВГ поставить на первое место букву Г. Например, можно поменять местами первую и последнюю буквы; получится слово ГЛМНБВК.

Полученный набор слов: {ГДЕЖЗИКЛ, ГЛМНБВК}. Теперь оба слова начинаются с буквы Г, и Петя должен написать Г. Вова пишет Д, получается ГД. Теперь в конце игры получится слово ГДЕЖЗИКЛ. Длина слова – 88 букв. Поэтому выигрывает Вова.

Задание 3

Выигрышная стратегия есть у Вовы. Петя первым ходом может поставить либо букву С, либо букву К. В зависимости от первой поставленной буквы, Вова следует действовать в соответствии со стратегией, изображенной на рисунке ниже. При любом ходе Пети у Вовы есть выигрышная стратегия. Дерево всех партий для этой стратегии показано на следующем рисунке. Подчёркнуты позиции, в которых партии заканчиваются.

1-й ход Пети 1-й ход Вовы 2-й ход Пети 2-й ход Вовы 3-й ход Пети 3-й ход Вовы

С —Т—> СТ —О—> СТО —Л—> СТОЛ

[пусто] —К—> КО —Л—> КОЛ —Е—> КОЛЕ —С—> КОЛЕС —О—> КОЛЕСО

Указания по оцениванию	Баллы
<i>Предварительные замечания</i> В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом	

соответствует количеству выполненных заданий (подробнее см. ниже).

Пункты 1А и 1Б считаются выполненными, если (I) правильно указано, кто из игроков имеет выигрышную стратегию в на данном наборе, и (II) описаны выигрышные стратегии – так, как это сделано в образце решения, или любым другим способом (таблица, словесно и т.д.).

Первое задание считается выполненным полностью, если выполнены полностью оба пункта: 1А и 1Б.

Замечание для проверяющего. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника (см. условие задачи). Экзаменуемый может описывать стратегию любым удобным ему способом. Существенно, чтобы было понятно, какой ход должен сделать игрок, реализующий стратегию, и было показано, что все возможные заключительные позиции выигрышные для этого игрока.

Задание 2 считается выполненным, если описана выигрышная стратегия Вовы.

Задание 3 считается выполненным, если правильно построено дерево всех возможных партий для выигрышной стратегии Вовы.

Если в дереве указаны пути, которые не могут встретиться при использовании выигрышной стратегии, то задание считается не выполненным. 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 описки, которые не искажают сути решения и не приводят к неправильному ответу	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. <ul style="list-style-type: none"> • Выполнено третье задание. • Выполнены первое и второе задания. 	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. <ul style="list-style-type: none"> • Первое задание выполнено, возможно, с недочётами, указанными в критериях на 3 балла. • Второе задание выполнено, возможно, с недочётами, указанными в критериях на 2 балла. 	1

Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
Максимальный балл	3

27 На вход программы поступает последовательность из NN целых положительных чисел. Известно, что все числа в последовательности различны. Необходимо составить программу, которая подсчитывает количество пар различных элементов последовательности, произведение которых делится на 51. Элементы пары не обязательно должны стоять в последовательности рядом, порядок элементов в паре не важен.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 10000$).

В каждой из последующих NN строк записано одно целое положительное число, не превышающее 10 000.

В качестве результата программа должна напечатать одно число: количество пар, в которых произведение элементов кратно 51.

Пример входных данных:

5
1
2
3
17
51

Пример выходных данных для приведённого выше примера входных данных:

5
Пояснение. Из пяти заданных чисел можно составить 10 попарных произведений:

1·2 1·3 1·17 1·51 2·3 2·17 2·51 3·17

Из них на 51 делятся 5 произведений

1·51 2·51 3·17 3·51 17·51

Требуется написать эффективную по времени и по памяти программу для решения описанной задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел N в k раз время работы программы увеличивается не более чем в k раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 килобайта и не увеличивается с ростом N .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, – 4 балла. Максимальная оценка за правильную программу, эффективную только по времени – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать одну программу или две программы решения задачи (например, одна из программ может быть менее эффективна). Если Вы сдадите две программы, то каждая из них будет оцениваться независимо от другой, итоговой станет большая из двух оценок.

Перед текстом программы обязательно кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

Содержание верного ответа и указания по оцениванию на 2 балла (допускаются иные формулировки ответа, не искажающие его смысла)

Описание алгоритма. Произведение двух чисел делится на 51, если выполнено одно из следующих условий (условия не могут выполняться одновременно).

- 1) Оба сомножителя делятся на 51
- 2) Один из сомножителей делится на 51, а другой не делится.
- 3) Ни один из сомножителей не делится на 51, но один сомножитель делится на 3, а другой – на 17.

При вводе данных можно определять, делится ли каждое из чисел на 51, 3 и 17, и подсчитывать следующие значения:

- $n51$ – количество чисел, кратных 51;
- $n17$ – количество чисел, кратных 17, но не кратных 51;
- $n3$ – количество чисел, кратных 3, но не кратных 51.

Сами числа при этом можно не хранить. Каждое число учитывается не более чем в одном из счётчиков.

Количество пар, удовлетворяющих условию 1), можно вычислить по формуле $n51 \cdot (n51 - 1) / 2$

Количество пар, удовлетворяющих условию 2), можно вычислить по формуле $n51 \cdot (N - n51)$

Количество пар, удовлетворяющих условию 3), можно вычислить по формуле $n3 \cdot n17$

Поэтому искомое количество пар вычисляется по формуле



$n51 \cdot (n51-1)/2 + n51 \cdot (N-n51) + n3 \cdot n17$
 Пример правильной программы на языке PascalABC с комментариями:

```

var
  N: integer; {общее количество чисел}
  a: integer; {очередное число}
  n51, n17, n3: integer; {счетчики чисел, кратных 51, 17, 3}
  k51: integer; {количество требуемых пар, ответ}
  i: integer;

begin
  readln(N);
  n51:=0; n17:=0; n3:=0;
  for i:=1 to N do begin
    readln(a);
    if a mod 51 = 0 then
      inc(n51)
    else if a mod 17 = 0 then
      inc(n17)
    else if a mod 3 = 0 then
      inc(n3);
  end;
  k51 := n51*(n51-1) div 2 + n51*(N-n51) + n3*n17;
  writeln(k51);
end.
    
```

- При таком решении каждое прочитанное число обрабатывается (делаются проверки делимости, изменяются счётчики) и после этого не хранится. Таким образом, используемая память не зависит от длины последовательности. Время обработки очередного числа фиксировано, т.е. не зависит от длины последовательности. Время заключительных вычислений по приведённой в решении формуле также не зависит от длины последовательности. Поэтому при увеличении длины последовательности в k раз время работы программы увеличивается не более чем в k раз. Таким образом, приведённая выше программа эффективна как по времени, так и по используемой памяти. Это решение оценивается 4 баллами.
- Общая идея решения, эффективного по времени, состоит в следующем. Просматриваем по очереди все элементы последовательности и накапливаем значения вспомогательных величин. После того как вся последовательность обработана и подсчитаны окончательные значения вспомогательных величин, по этим значениям подсчитывается искомое количество пар. При этом

можно использовать и другие вспомогательные величины. Все подобные программы оцениваются в 4 балла.

- Возможно решение, основанное на описанных идеях, однако предварительно сохраняющее элементы последовательности в массив. Такое решение (если в нём нет ошибок) эффективно по времени, но неэффективно по памяти. Оно оценивается в 3 балла.
- Решение, не эффективное ни по времени, ни по памяти, запоминает входную последовательность в массиве, после чего явно перебирает все возможные пары. Такое решение оценивается в 2 балла (см. критерий).

Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается большая из двух оценок. Описание алгоритма решения не оценивается.

Указания по оцениванию	Баллы
<p>Предварительные замечания.</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. Случай 2.1. Ученик представил только одну программу. 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. Следует рассматривать вторую программу как ответ на оставшееся задание.</p> <p>Случай 2.3. Ученик представил две программы; ни для одной из них задание не указано, или в обоих решениях указано одно и то же задание. Следует первую (по порядку в представленных учеником материалах) программу рассматривать как ответ на задание А, а вторую – как ответ на задание Б.</p> <p>Случай 2.4. Ученик представил более двух программ. Следует рассматривать только две последние программы и соотносить их с заданиями по правилам 2.1–2.3.</p>	





<p>Случай 2.5. Решение, представленное в качестве решения задания А, по критериям для задания Б может быть оценено в 3 или 4 балла. При этом решение, представленное в качестве решения задания Б, оценено меньшим баллом. Следует считать, что ученик перепутал обозначения заданий и оценивать решение, представленное как решение задания А, по критериям задания Б.</p> <p><i>НАПОМИНАЕМ! Итоговый балл за задачу – это больший из баллов, полученных учеником за каждое из двух представленных решений.</i></p> <p><i>Пояснения для проверяющих.</i></p> <p>1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то считается, что учеником допущена опечатка, и решение оценивается по критериям для задания А. В качестве решения задания А может быть представлена программа, которая решает задачу при произвольном количестве входных данных (как в задании Б), осуществляя полный перебор всех вариантов. Такая программа неэффективна по времени, поэтому она должна оцениваться 2 баллами.</p> <p>2. Два задания (и, соответственно, возможность для экзаменуемого представить две программы) дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое даёт ему право получить 2 балла, а затем приступить к поиску более эффективного решения.</p> <p>3. Приведённые в п. 2.1–2.5 правила имеют целью избежать снижения баллов из-за того, что ученик перепутал обозначения заданий.</p> <p><i>Общие принципы оценивания решений</i></p> <p>4 балла ставится за эффективную и правильно работающую программу, решающую задачу в общем случае (задача Б). При этом программа может содержать до трёх синтаксических ошибок («описок»).</p> <p>3 балла ставится в случае, когда фактически задача решена и решена эффективно по времени, возможно, с хранением всех входных данных в массиве, но количество «описок» более трёх (но не более пяти) и допущено не более одной</p>	
--	--

<p>содержательной ошибки, не позволяющей усомниться в том, что экзаменуемый правильно придумал алгоритм.</p> <p>2 балла ставится, если программа, решающая задачу Б, в дополнение к неточностям, которые перечислены выше, работает неэффективно по времени и/или допущено до трёх упомянутых выше содержательных ошибок. Количество допустимых «описок» – до семи.</p> <p>2 балла также ставится за правильное решение упрощенной задачи (задача А).</p> <p>1 балл ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Далее эти общие принципы уточнены</p>	
Критерии оценивания задания А	
<p>При решении задачи А программа верно находит требуемую сумму для любых 6 пар исходных данных. Допускается до пяти синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла)</p>	2
<p>Не выполнены условия, позволяющие поставить 2 балла. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. Допускается любое количество «описок»</p>	1
<p>Не выполнены критерии, позволяющие поставить 1 или 2 балла</p>	0
<i>Максимальный балл для задания А</i>	2
Критерии оценивания задания Б	
<p>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 (в том числе стек рекурсивных вызовов), размер которых зависит от количества входных элементов, а время работы пропорционально этому количеству. Возможно использование массивов и динамических структур данных при условии, что в них в каждый момент времени хранится фиксированное количество элементов, требующих для хранения меньше 1Кб.</p>	4

<p>Программа может содержать не более трёх синтаксических ошибок следующих видов:</p> <ol style="list-style-type: none"> 1) пропущен или неверно указан знак пунктуации; 2) неверно написано или пропущено зарезервированное слово языка программирования; 3) не описана или неверно описана переменная; 4) применяется операция, недопустимая для соответствующего типа данных. <p>К синтаксическим ошибкам приравнивается использование неверного типа данных. Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку</p>	
<p>Не выполнены условия, позволяющие поставить 4 балла. Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел; правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел. Количество синтаксических ошибок («описок») указанных выше видов – не более пяти.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в C++ или другой структуре данных). Допускается ошибка при вводе и выводе данных, не влияющая на содержание решения.</p> <p>Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла. Кроме того, допускается наличие одной ошибки, принадлежащей к одному из следующих видов:</p> <ol style="list-style-type: none"> 1) ошибка инициализации, в том числе отсутствие инициализации; 2) не выводится результат, равный 0, или вместо 0 выводится неверное значение; 3) допущен выход за границу массива; 4) используется знак “<” вместо “<=”, “or” вместо “and” и т.п. 	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет, например, для решения задачи используется перебор всех возможных вариантов выбора элементов в парах. В реализации алгоритма допускается до трёх содержательных</p>	2

<p>ошибок, допустимые виды ошибок перечислены в критериях на 3 балла. Количество синтаксических «описок» не должно быть более семи. Программа может быть неэффективна по времени, например, все числа запоминаются в массиве и перебираются все возможные суммы, т.е., по сути, реализовано решение задачи А без ограничений на количество введённых пар</p>	
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. Допускается любое количество «описок». 1 балл ставится также за решения, верные лишь в частных случаях</p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p><i>Максимальный балл для задания Б</i></p>	4
<p><i>Итоговый максимальный балл</i></p>	4

