

Проект

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

Демонстрационный вариант
контрольных измерительных материалов единого
государственного экзамена 2018 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 2018 года по ИНФОРМАТИКЕ и ИКТ

При ознакомлении с демонстрационным вариантом контрольных измерительных материалов ЕГЭ 2018 г. следует иметь в виду, что задания, включённые в него, не отражают всех вопросов содержания, которые будут проверяться с помощью вариантов КИМ в 2018 г. Полный перечень вопросов, которые могут контролироваться на едином государственном экзамене 2018 г., приведён в кодификаторе элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения единого государственного экзамена 2018 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.

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 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.

1 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 ручки.

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 существует целых чисел x , для которых выполняется неравенство $2A_{16} < x < 61_8$?
В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____.

2 Логическая функция F задаётся выражением $\neg x \vee y \vee (\neg z \wedge w)$.
На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F ложна. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных w, x, y, z .

| Переменная 1 | Переменная 2 | Переменная 3 | Переменная 4 | Функция |
|--------------|--------------|--------------|--------------|---------|
| ??? | ??? | ??? | ??? | F |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |

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: x и y , и был приведён фрагмент её таблицы истинности, содержащий все наборы аргументов, при которых функция истинна.

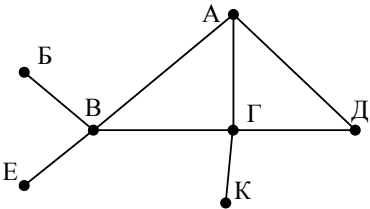
| Переменная 1 | Переменная 2 | Функция |
|--------------|--------------|---------|
| ??? | ??? | F |
| 0 | 0 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Тогда первому столбцу соответствовала бы переменная y , а второму столбцу – переменная x . В ответе следовало бы написать: yx .

Ответ: _____.

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

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|----|----|----|----|----|----|----|----|
| П1 | | 7 | | | | | |
| П2 | 7 | | 8 | | 3 | 4 | |
| П3 | | 8 | | 11 | 6 | | |
| П4 | | | 11 | | 5 | | |
| П5 | | 3 | 6 | 5 | | | 9 |
| П6 | | 4 | | | | | |
| П7 | | | | | 9 | | |



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 протяжённость дороги из пункта А в пункт Г. В ответе запишите целое число – так, как оно указано в таблице.

Ответ: _____.

- 4 Ниже представлены два фрагмента таблиц из базы данных о жителях микрорайона. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных, у скольких детей на момент их рождения матерям было больше 22 полных лет. При вычислении ответа учитывайте только информацию из приведённых фрагментов таблиц.

| Таблица 1 | | | | Таблица 2 | |
|-----------|---------------|-----|--------------|-------------|------------|
| ID | Фамилия_И.О. | Пол | Год_рождения | ID_Родителя | ID_Ребёнка |
| 15 | Петрова Н.А. | Ж | 1944 | 22 | 23 |
| 22 | Иваненко И.М. | М | 1940 | 42 | 23 |
| 23 | Иваненко М.И. | М | 1968 | 23 | 24 |
| 24 | Иваненко М.М. | М | 1993 | 73 | 24 |
| 32 | Будай А.И. | Ж | 1960 | 22 | 32 |
| 33 | Будай В.С. | Ж | 1987 | 42 | 32 |
| 35 | Будай С.С. | М | 1965 | 32 | 33 |
| 42 | Коладзе А.С. | Ж | 1941 | 35 | 33 |
| 43 | Коладзе Л.А. | М | 1955 | 15 | 35 |
| 44 | Родэ О.С. | М | 1990 | 32 | 44 |
| 46 | Родэ М.О. | М | 2010 | 35 | 44 |
| 52 | Ауэрман А.М. | Ж | 1995 | 23 | 52 |
| 73 | Антонова М.А. | Ж | 1967 | 73 | 52 |
| ... | ... | ... | ... | ... | ... |

Ответ: _____.

- 5 По каналу связи передаются шифрованные сообщения, содержащие только десять букв: А, Б, Е, И, К, Л, Р, С, Т, У. Для передачи используется неравномерный двоичный код. Для девяти букв используются кодовые слова.

| Буква | Кодовое слово | Буква | Кодовое слово |
|-------|---------------|-------|---------------|
| А | 00 | Л | 1101 |
| Б | | Р | 1010 |
| Е | 010 | С | 1110 |
| И | 011 | Т | 1011 |
| К | 1111 | У | 100 |

Укажите кратчайшее кодовое слово для буквы Б, при котором код будет удовлетворять условию Фано. Если таких кодов несколько, укажите код с **наименьшим** числовым значением.

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

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 числа N , 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 её 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 минимальное число R , которое превышает число 83 и может являться результатом работы данного алгоритма. В ответе это число запишите в десятичной системе счисления.

Ответ: _____.

- 7 Дан фрагмент электронной таблицы. Из ячейки B3 в ячейку A4 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Каким стало числовое значение формулы в ячейке A4?

| | A | B | C | D | E |
|---|---|---------------|-----|------|-------|
| 1 | 1 | 10 | 100 | 1000 | 10000 |
| 2 | 2 | 20 | 200 | 2000 | 20000 |
| 3 | 3 | = \$C2 + D\$3 | 300 | 3000 | 30000 |
| 4 | | 40 | 400 | 4000 | 40000 |

Примечание: знак \$ обозначает абсолютную адресацию.

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 = 260 N = 0 WHILE S > 0 S = S - 15 N = N + 2 WEND PRINT N </pre> | <pre> s = 260 n = 0 while s > 0: s = s - 15 n = n + 2 print(n) </pre> |
| Алгоритмический язык | Паскаль |
| <pre> алг нач цел n, s s := 260 n := 0 нц пока s > 0 s := s - 15 n := n + 2 кц вывод n кон </pre> | <pre> var s, n: integer; begin s := 260; n := 0; while s > 0 do begin s := s - 15; n := n + 2; end; writeln(n); end. </pre> |

```

C++
#include <iostream>
using namespace std;

int main() {
    int s = 260, n = 0;
    while (s > 0) {
        s = s - 15;
        n = n + 2;
    }
    cout << n << endl;
    return 0;
}
        
```

Ответ: _____.

- 9 Автоматическая фотокамера производит растровые изображения размером 640×480 пикселей. При этом объём файла с изображением не может превышать 320 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре?

Ответ: _____.

- 10 Все 4-буквенные слова, составленные из букв Д, Е, К, О, Р, записаны в алфавитном порядке и пронумерованы, начиная с 1.

Ниже приведено начало списка.

1. ДДДД
2. ДДДЕ
3. ДДДК
4. ДДДО
5. ДДДР
6. ДДЕД

...

Под каким номером в списке идёт первое слово, которое начинается с буквы К?

Ответ: _____.

11 Ниже на пяти языках программирования записан рекурсивный алгоритм F.

| Бейсик | Python |
|---|--|
| <pre>SUB F(n) IF n > 0 THEN PRINT n F(n - 3) F(n \ 3) END IF END SUB</pre> | <pre>def F(n): if n > 0: print(n) F(n - 3) F(n // 3)</pre> |
| Алгоритмический язык | Паскаль |
| <pre>алг F(цел n) нач если n > 0 то вывод n F(n - 3) F(div(n, 3)) все кон</pre> | <pre>procedure F(n: integer); begin if n > 0 then begin write(n); F(n - 3); F(n div 3) end end;</pre> |
| C++ | |
| <pre>void F(int n){ if (n > 0){ std::cout <<n; F(n - 3); F(n / 3); } }</pre> | |

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

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 57.179.208.27 адрес сети равен 57.179.192.0. Каково наибольшее возможное количество единиц в разрядах маски?

Ответ: _____.

13 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 10 символов. В качестве символов используют прописные буквы латинского алфавита, т.е. 26 различных символов. В базе данных для хранения каждого пароля отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Определите объём памяти (в байтах), необходимый для хранения данных о 50 пользователях. В ответе запишите только целое число – количество байт.

Ответ: _____.

- 14** Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **сместиться на (a, b)** , где a, b – целые числа. Эта команда перемещает Чертёжника из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Например, если Чертёжник находится в точке с координатами $(4, 2)$, то команда **сместиться на $(2, -3)$** переместит Чертёжника в точку $(6, -1)$.

Цикл

ПОВТОРИ число РАЗ
последовательность команд
КОНЕЦ ПОВТОРИ

означает, что последовательность команд будет выполнена указанное число раз (число должно быть натуральным).

Чертёжнику был дан для исполнения следующий алгоритм (число повторений и величины смещения в первой из повторяемых команд неизвестны):

НАЧАЛО

сместиться на $(4, 6)$

ПОВТОРИ ...РАЗ

сместиться на $(..., ...)$

сместиться на $(4, -6)$

КОНЕЦ ПОВТОРИ

сместиться на $(-28, -22)$

КОНЕЦ

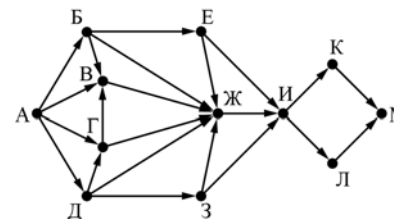
В результате выполнения этого алгоритма Чертёжник возвращается в исходную точку. Какое наибольшее число повторений могло быть указано в конструкции «ПОВТОРИ ... РАЗ»?

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, проходящих через город Ж?



Ответ: _____.

- 16** Значение арифметического выражения: $49^{10} + 7^{30} - 49$ – записали в системе счисления с основанием 7. Сколько цифр «6» 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 сотнях тысяч) |
|-------------------------------------|-------------------------------------|
| <i>Бабочка</i> | 22 |
| <i>Гусеница</i> | 40 |
| <i>Трактор</i> | 24 |
| <i>Трактор Бабочка Гусеница</i> | 66 |
| <i>Трактор & Гусеница</i> | 12 |
| <i>Трактор & Бабочка</i> | 0 |

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Бабочка & Гусеница*?

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

Ответ: _____.

- 18** Для какого наибольшего целого числа A формула

$$((x \leq 9) \rightarrow (x \cdot x \leq A)) \wedge ((y \cdot y \leq A) \rightarrow (y \leq 9))$$

тождественно истинна, то есть принимает значение 1 при любых целых неотрицательных x и y ?

Ответ: _____.

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

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

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

Ответ: _____.

20

Ниже на пяти языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: L и M . Укажите **наименьшее** число x , при вводе которого алгоритм печатает сначала 5, а потом 7.

| Бейсик | Python |
|--|--|
| <pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X > 0 M = M + 1 IF X MOD 2 <> 0 THEN L = L + 1 END IF X = X \ 2 WEND PRINT L PRINT M </pre> | <pre> x = int(input()) L = 0 M = 0 while x > 0: M = M + 1 if x % 2 != 0: L = L + 1 x = x // 2 print(L) print(M) </pre> |
| Алгоритмический язык | Паскаль |
| <pre> алг нач цел x, L, M <u>ВВОД</u> x L := 0 M := 0 <u>нц пока</u> x > 0 M := M + 1 <u>если</u> mod(x,2) <> 0 <u>то</u> L := L + 1 <u>все</u> x := div(x,2) <u>кц</u> <u>ВЫВОД</u> L, <u>нс</u>, M кон </pre> | <pre> var x, L, M: integer; begin readln(x); L := 0; M := 0; while x>0 do begin M := M + 1; if x mod 2 <> 0 then L := L + 1; x := x div 2; end; writeln(L); writeln(M); end. </pre> |

C++

```

#include <iostream>
using namespace std;

int main(){
    int x, L, M;
    cin >> x;
    L = 0;
    M = 0;
    while (x > 0) {
        M = M + 1;
        if(x % 2 != 0) {
            L = L + 1;
        }
        x = x / 2;
    }
    cout << L << endl << M << endl;
    return 0;
}

```

Ответ: _____.

21

Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма. Для Вашего удобства алгоритм представлен на пяти языках программирования.

| Бейсик | Python |
|---|--|
| <pre> DIM A, B, T, M, R AS LONG A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) <= R THEN M = T R = F(T) END IF NEXT T PRINT M+R FUNCTION F (x) F = 2*(x*x-1)*(x*x-1)+27 END FUNCTION </pre> | <pre> def F(x): return 2*(x*x-1)*(x*x-1)+27 a = -20; b=20 M=a; R=F(a) for t in range(a,b+1): if (F(t) <= R): M=t; R=F(t) print (M+R) </pre> |

| Алгоритмический язык | Паскаль |
|---|---|
| <pre> алг нач цел a, b, t, M, R a:=-20; b:=20 M:=a; R:=F(a) нц для t от a до b если F(t) <= R то M:=t; R:=F(t) все кц вывод M+R кон алг цел F(цел x) нач знач:=2*(x*x-1)*(x*x-1)+27 кон </pre> | <pre> var a, b, t, M, R :longint; function F(x: longint) : longint; begin F:= 2*(x*x-1)*(x*x-1)+27; end; begin a:=-20; b:=20; M:=a; R:=F(a); for t:= a to b do begin if (F(t) <= R) then begin M:=t; R:=F(t) end; end; write (M+R) end. </pre> |
| C++ <pre> #include <iostream> using namespace std; long F(long x) { return 2*(x*x-1)*(x*x-1)+27; } int main() { long a = -20, b = 20, M = a, R = F(a); for (int t = a; t <= b; ++t) { if (F(t)<= R) { M = t; R = F(t); } } cout << M + R; return 0; } </pre> | |

Ответ: _____.

22

Исполнитель М17 преобразует число, записанное на экране. У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1
2. Прибавить 2
3. Умножить на 3

Первая из них увеличивает число на экране на 1, вторая увеличивает его на 2, третья умножает на 3.

Программа для исполнителя М17 – это последовательность команд.

Сколько существует таких программ, которые преобразуют исходное число 2 в число 12 и при этом траектория вычислений программы содержит числа 8

и 10? Траектория должна содержать оба указанных числа.

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **132** при исходном числе 7 траектория будет состоять из чисел 8, 24, 26.

Ответ: _____.

23

Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$, которые удовлетворяют всем перечисленным ниже условиям?

$$(\neg x_1 \vee y_1) \rightarrow (\neg x_2 \wedge y_2) = 1$$

$$(\neg x_2 \vee y_2) \rightarrow (\neg x_3 \wedge y_3) = 1$$

...

$$(\neg x_6 \vee y_6) \rightarrow (\neg x_7 \wedge y_7) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$, 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 натуральное число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 5. Если в числе нет цифр, кратных 5, требуется на экран вывести «NO». Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Напоминание: 0 делится на любое натуральное число.

| Бейсик | Python |
|---|--|
| <pre> DIM N, DIGIT, MAXDIGIT AS LONG INPUT N MAXDIGIT = N MOD 10 WHILE N > 0 DIGIT = N MOD 10 IF DIGIT MOD 5 = 0 THEN IF DIGIT > MAXDIGIT THEN MAXDIGIT = DIGIT END IF END IF N = N \ 10 WEND IF MAXDIGIT = 0 THEN PRINT "NO" ELSE PRINT MAXDIGIT END IF </pre> | <pre> N = int(input()) maxDigit = N % 10 while N > 0: digit = N % 10 if digit % 5 == 0: if digit > maxDigit: maxDigit = digit N = N // 10 if maxDigit == 0: print("NO") else: print(maxDigit) </pre> |

| Алгоритмический язык | Паскаль |
|---|---|
| <pre> алг нач цел N, digit, maxDigit ввод N maxDigit := mod(N,10) нц пока N > 0 digit := mod(N,10) если mod(digit, 5) = 0 то если digit > maxDigit то maxDigit := digit все N := div(N,10) кц если maxDigit = 0 то вывод "NO" иначе вывод maxDigit все кон </pre> | <pre> var N,digit,maxDigit: longint; begin readln(N); maxDigit := N mod 10; while N > 0 do begin digit := N mod 10; if digit mod 5 = 0 then if digit > maxDigit then maxDigit := digit; N := N div 10; end; end if maxDigit = 0 then writeln('NO') else writeln(maxDigit) end. </pre> |
| C++ | |
| <pre> #include <iostream> using namespace std; int main() { long N, digit, maxDigit; cin >> N; maxDigit = N % 10; while (N > 0) { digit = N % 10; if (digit % 5 == 0) if (digit > maxDigit) maxDigit = digit; N = N / 10; } if (maxDigit == 0) cout << "NO" << endl; else cout << maxDigit << endl; return 0; } </pre> | |

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 132.
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 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит количество элементов массива, больших 100 и при этом кратных 5, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строчки.

Например, для массива из шести элементов: 4 115 7 195 25 106 программа должна вывести числа 4 2 7 2 25 106

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.

| Бейсик | Python |
|---|---|
| <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 ...</pre> | <pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n): a.append(int(input())) ...</pre> |
| END | |

| Алгоритмический язык | Паскаль |
|---|--|
| <pre>алг нач цел N = 30 цел таб a[1:N] цел i, j, k нц для i от 1 до N ввод a[i] кц ... кон</pre> | <pre>const N = 30; var a: array [1..N] of longint; i, j, k: longint; begin for i := 1 to N do readln(a[i]); ... end.</pre> |
| C++ | |
| <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> | |

В качестве ответа Вам необходимо привести фрагмент программы, 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). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).

26 Два игрока, Петя и Ваня играют в следующую игру. На столе в кучке лежат фишки. На лицевой стороне каждой фишки написано двузначное натуральное число, обе цифры которого находятся в диапазоне от 1 до 4. Никакие две фишки не повторяются. Игра состоит в том, что игроки поочередно берут из кучки по одной фишке и выкладывают в цепочку на стол лицевой стороной вверх таким образом, что каждая новая фишка ставится **правее** предыдущей и ближайшие цифры соседних фишек совпадают. Верхняя часть всех выложенных фишек направлена в одну сторону, то есть переворачивать фишки нельзя. Например, из фишки, на которой написано 23, нельзя сделать фишку, на которой написано 32. Первый ход делает Петя, выкладывая на стол любую фишку из кучки. Игра заканчивается, когда в кучке нет ни одной фишки, которую можно добавить

в цепочку. Тот, кто добавил в цепочку последнюю фишку, **выигрывает**, а его противник проигрывает.

Будем называть **партией** любую допустимую правилами последовательность ходов игроков, приводящую к завершению игры.

Будем говорить, что игрок имеет **выигрышную стратегию**, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит указать, какую фишку он должен выставить в любой ситуации, которая ему может встретиться при различной игре противника.

Пример партии.

Пусть на столе в кучке лежат фишки: 11, 12, 13, 21, 22, 23

Пусть первый ход Пети 12.

Ваня может поставить 21, 22 или 23. Предположим, он ставит 21. Получим цепочку 12-21.

Петя может поставить 11 или 13. Предположим, он ставит 11. Получим цепочку 12-21-11.

Ваня может поставить только фишку со значением 13. Получим цепочку 12-21-11-13.

Перед Петей в кучке остались только фишки 22 и 23, то есть нет фишек, которые он мог бы добавить в цепочку. Таким образом, партия закончена, Ваня выиграл.

Выполните следующие три задания при исходном наборе фишек в кучке {12, 14, 21, 22, 24, 41, 42, 44}.

Задание 1.

Приведите пример самой короткой партии, возможной при данном наборе фишек. Если таких партий несколько, достаточно привести одну.

Задание 2. Пусть Петя первым ходом пошел 44. У кого из игроков есть выигрышная стратегия, позволяющая в этой ситуации выиграть своим четвертым ходом? Постройте в виде рисунка или таблицы дерево всех партий, возможных при реализации выигрывающим игроком этой стратегии. На рёбрах дерева указывайте ход, в узлах – цепочку фишек, получившуюся после этого хода.

Задание 3. Укажите хотя бы один способ убрать 2 фишки из исходного набора так, чтобы всегда выигрывал **не** тот игрок, который имеет выигрышную стратегию в задании 2. Приведите пример партии для набора из 6 оставшихся фишек.

27 На вход программы поступает последовательность из N целых положительных чисел, все числа в последовательности различны. Рассматриваются все пары различных элементов последовательности (элементы пары не обязаны стоять в последовательности рядом, порядок элементов в паре не важен). Необходимо определить количество пар, для которых произведение элементов делится на 26.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$). В каждой из последующих N 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 26.

Пример входных данных:

4
2
6
13
39

Пример выходных данных для приведённого выше примера входных данных:

4

Пояснение. Из четырёх заданных чисел можно составить 6 попарных произведений: $2 \cdot 6$, $2 \cdot 13$, $2 \cdot 39$, $6 \cdot 13$, $6 \cdot 39$, $13 \cdot 39$ (результаты: 12, 26, 78, 78, 234, 507). Из них на 26 делятся 4 произведения ($2 \cdot 13 = 26$; $2 \cdot 39 = 78$; $6 \cdot 13 = 78$; $6 \cdot 39 = 234$).

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 Кбайт 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.

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 | xzwy |
| 3 | 6 |
| 4 | 5 |
| 5 | 1100 |
| 6 | 86 |
| 7 | 600 |
| 8 | 36 |
| 9 | 256 |
| 10 | 251 |
| 11 | 9631231 |
| 12 | 19 |
| 13 | 350 |
| 14 | 8 |
| 15 | 20 |
| 16 | 18 |
| 17 | 8 |
| 18 | 99 |
| 19 | 5 |
| 20 | 79 |
| 21 | 28 |
| 22 | 60 |
| 23 | 22 |

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 натуральное число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 5. Если в числе нет цифр, кратных 5, требуется на экран вывести «NO». Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Напоминание: 0 делится на любое натуральное число.

| Бейсик | Python |
|---|--|
| <pre>DIM N, DIGIT, MAXDIGIT AS LONG INPUT N MAXDIGIT = N MOD 10 WHILE N > 0 DIGIT = N MOD 10 IF DIGIT MOD 5 = 0 THEN IF DIGIT > MAXDIGIT THEN MAXDIGIT = DIGIT END IF END IF N = N \ 10 WEND IF MAXDIGIT = 0 THEN PRINT "NO" ELSE PRINT MAXDIGIT END IF</pre> | <pre>N = int(input()) maxDigit = N % 10 while N > 0: digit = N % 10 if digit % 5 == 0: if digit > maxDigit: maxDigit = digit N = N // 10 if maxDigit == 0: print("NO") else: print(maxDigit)</pre> |

| Алгоритмический язык | Паскаль |
|---|--|
| <pre> алг нач цел N, digit, maxDigit ввод N maxDigit := mod(N,10) нц пока N > 0 digit := mod(N,10) если mod(digit, 5) = 0 то если digit > maxDigit то maxDigit := digit все все N := div(N,10) кц если maxDigit = 0 то вывод "NO" иначе вывод maxDigit все кон </pre> | <pre> var N,digit,maxDigit: longint; begin readln(N); maxDigit := N mod 10; while N > 0 do begin digit := N mod 10; if digit mod 5 = 0 then if digit > maxDigit then maxDigit := digit; N := N div 10 end; if maxDigit = 0 then writeln('NO') else writeln(maxDigit) end. end. end. </pre> |
| C++ | |
| <pre> #include <iostream> using namespace std; int main() { long N, digit, maxDigit; cin >> N; maxDigit = N % 10; while (N > 0) { digit = N % 10; if (digit % 5 == 0) if (digit > maxDigit) maxDigit = digit; N = N / 10; } if (maxDigit == 0) cout << "NO" << endl; else cout << maxDigit << endl; return 0; } </pre> | |

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 132.
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) |
|---|
| <p>Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков.</p> <p>1. Программа выведет число 2.</p> <p>2. Программа выдаёт правильный ответ, например, для числа 135.</p> <p><i>Замечание для проверяющего. Программа работает неправильно из-за неверной начальной инициализации и неверной проверки отсутствия цифр, кратных 5. Соответственно, программа будет выдавать верный ответ, если вводимое число содержит хотя бы одну цифру, кратную 5, и наибольшая цифра числа, кратная 5, не равна 0 и не меньше младшей (крайней правой) цифры числа (или просто стоит последней).</i></p> <p>3. В программе есть две ошибки.</p> <p>Первая ошибка. Неверная инициализация ответа (переменная maxDigit).</p> <p>Строка с ошибкой: maxDigit := N mod 10;</p> <p>Верное исправление: maxDigit := -1;</p> <p>Вместо -1 может быть использовано любое число, меньшее 0.</p> <p>Вторая ошибка. Неверная проверка отсутствия цифр, кратных 5.</p> <p>Строка с ошибкой: if maxDigit = 0 then</p> <p>Верное исправление: if maxDigit = -1 then</p> <p>Вместо -1 может быть другое число, меньшее 0, которое было положено в maxDigit при исправлении первой ошибки, или проверка, что maxDigit < 0</p> |

| Указания по оцениванию | Баллы |
|--|-------|
| <p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <p>1) указать, что выведет программа при конкретном входном числе;</p> <p>2) указать пример входного числа, при котором программа выдаёт верный ответ;</p> <p>3) исправить первую ошибку;</p> <p>4) исправить вторую ошибку.</p> <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p> <p>Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия:</p> <p>а) правильно указана строка с ошибкой;</p> <p>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</p> | |
| Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной | 3 |
| <p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <p>а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной;</p> <p>б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки</p> | 2 |
| Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх | 1 |
| Не выполнены условия, позволяющие поставить 1, 2 или 3 балла | 0 |
| Максимальный балл | 3 |

25

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит количество элементов массива, больших 100 и при этом кратных 5, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строки.

Например, для массива из шести элементов: 4 115 7 195 25 106
программа должна вывести числа 4 2 7 2 25 106

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.

| Бейсик | Python |
|---|---|
| <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 ...</pre> | <pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n): a.append(int(input())) ...</pre> |
| END | |

| Алгоритмический язык | Паскаль |
|---|--|
| <u>алг</u> <u>нач</u> цел N = 30 целтаб a[1:N] цел i, j, k нц для i от 1 до N ввод a[i] кц ... <u>кон</u> | 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. |
| C++ | |
| #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; } | |

В качестве ответа Вам необходимо привести фрагмент программы, 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). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).

| Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла) |
|---|
| На языке Паскаль |
| <pre> k := 0; for i := 1 to N do if (a[i] > 100) and (a[i] mod 5 = 0) then k:=k+1; for i := 1 to N do begin if (a[i] > 100) and (a[i] mod 5 = 0) then a[i] := k; writeln(a[i]) end </pre> |

| На Алгоритмическом языке |
|--|
| <pre> k := 0 нц для i от 1 до N если a[i] > 100 и mod(a[i], 5) = 0 то k := k + 1 все кц нц для i от 1 до N если a[i] > 100 и mod(a[i], 5) = 0 то a[i] := k все вывод a[i], нс кц </pre> |
| На языке Бейсик |
| <pre> K = 0 FOR I = 1 TO N IF A(I) > 100 AND A(I) MOD 5 = 0 THEN K = K + 1 END IF NEXT I FOR I = 1 TO N IF A(I) > 100 AND A(I) MOD 5 = 0 THEN A(I) = K END IF PRINT A(I) NEXT I </pre> |
| На языке C++ |
| <pre> k = 0; for (i = 0; i < N; i++) if (a[i] > 100 && a[i]%5 == 0) k++; for (i = 0; i < N; i++) { if (a[i] > 100 && a[i]%5 == 0) a[i] = k; cout << a[i]; } </pre> |
| На языке Python |
| <pre> k = 0 for i in range(0, n): if (a[i] > 100 and a[i] % 5 == 0): k = k + 1 for i in range(0, n): if (a[i] > 100 and a[i] % 5 == 0): a[i] = k print(a[i]) </pre> |

| 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, приведённых 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 Алгоритмическом 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> <p>4. Допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</p> <p>5. Допускается формат вывода массива, отличный от указанного, например, в строку.</p> | |
| Предложен правильный алгоритм, выдающий в качестве результата верное значение | 2 |
| <p>Не выполнены условия, позволяющие поставить 2 балла. При этом предложено в целом верное решение, содержащее не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1) в цикле происходит выход за границу массива; 2) не инициализируется или неверно инициализируется количество найденных элементов; 3) неверно осуществляется проверка делимости на 5; 4) проверяется делимость на 5 не элемента массива, а его индекса; 5) неверно осуществляется сравнение с 100 (например, используется знак "больше или равно"); 6) сравнение с 100 производится для индекса массива, а не для его значения; 7) неверно составлено логическое условие (например, используется <code>or</code> вместо <code>and</code>); 8) не вычисляется или неверно вычисляется количество найденных элементов; 9) исходный массив не изменяется; 10) отсутствует вывод ответа или ответ выводится не полностью (например, только один элемент массива ввиду пропущенного цикла или операторных скобок); | 1 |

| | |
|--|---|
| <p>11) используется переменная, не объявленная в разделе описания переменных;</p> <p>12) не указано или неверно указано условие завершения цикла;</p> <p>13) индексная переменная в цикле не меняется (например, в цикле <code>while</code>) или меняется неверно.</p> | |
| Ошибок, перечисленных в п. 1–13, 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. На столе в кучке лежат фишки. На лицевой стороне каждой фишки написано двузначное натуральное число, обе цифры которого находятся в диапазоне от 1 до 4. Никакие две фишки не повторяются. Игра состоит в том, что игроки поочередно берут из кучки по одной фишке и выкладывают в цепочку на стол лицевой стороной вверх таким образом, что каждая новая фишка ставится **правее** предыдущей и ближайшие цифры соседних фишек совпадают. Верхняя часть всех выложенных фишек направлена в одну сторону, то есть переворачивать фишки нельзя. Например, из фишки, на которой написано 23, нельзя сделать фишку, на которой написано 32.

Первый ход делает Петя, выкладывая на стол любую фишку из кучки. Игра заканчивается, когда в кучке нет ни одной фишки, которую можно добавить в цепочку. Тот, кто добавил в цепочку последнюю фишку, **выигрывает**, а его противник проигрывает.

Будем называть **партией** любую допустимую правилами последовательность ходов игроков, приводящую к завершению игры.

Будем говорить, что игрок имеет **выигрышную стратегию**, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит указать, какую фишку он должен выставить в любой ситуации, которая ему может встретиться при различной игре противника.

Пример партии.

Пусть на столе в кучке лежат фишки: 11, 12, 13, 21, 22, 23

Пусть первый ход Пети 12.

Ваня может поставить 21, 22 или 23. Предположим, он ставит 21. Получим цепочку 12-21.

Петя может поставить 11 или 13. Предположим, он ставит 11. Получим цепочку 12-21-11.

Ваня может поставить только фишку со значением 13. Получим цепочку 12-21-11-13.

Перед Петей в кучке остались только фишки 22 и 23, то есть нет фишек, которые он мог бы добавить в цепочку. Таким образом, партия закончена, Ваня выиграл.

Выполните следующие три задания при исходном наборе фишек в кучке {12, 14, 21, 22, 24, 41, 42, 44}.

Задание 1.

Приведите пример самой короткой партии, возможной при данном наборе фишек. Если таких партий несколько, достаточно привести одну.

Задание 2. Пусть Петя первым ходом пошел 44. У кого из игроков есть выигрышная стратегия, позволяющая в этой ситуации выиграть своим четвертым ходом? Постройте в виде рисунка или таблицы дерево всех партий, возможных при реализации выигрывающим игроком этой стратегии. На рёбрах дерева указывайте ход, в узлах – цепочку фишек, получившуюся после этого хода.

Задание 3. Укажите хотя бы один способ убрать 2 фишки из исходного набора так, чтобы всегда выигрывал не тот игрок, который имеет выигрышную стратегию в задании 2. Приведите пример партии для набора из 6 оставшихся фишек.

| Содержание верного ответа | | | |
|---|---------------------------|----|----|
| (допускаются иные формулировки ответа, не искажающие его смысла) | | | |
| Задание 1. | | | |
| Все кратчайшие партии: 12-21-14-41, 14-41-12-21. | | | |
| Экзаменуемому достаточно указать одну из них. | | | |
| Задание 2. | | | |
| Выигрышная стратегия, позволяющая выиграть своим четвертым ходом, есть у Пети. Дерево всех партий для этой стратегии показано в таблице 1 и на рисунке 1. | | | |
| <i>Примечание для проверяющего.</i> Оба способа изображения допустимы. Ученику достаточно привести один из них. | | | |
| | Начальная позиция [пусто] | | |
| 1-й ход Пети | 44 | | |
| 1-й ход Вани | 41 | 42 | |
| 2-й ход Пети | 14 | 21 | |
| 2-й ход Вани | 42 | 12 | 14 |
| 3-й ход Пети | 21 | 24 | 41 |
| 3-й ход Вани | 12 | 41 | 12 |
| 4-й ход Пети | 24 | 14 | 24 |

Таблица 1. Дерево всех партий при выигрышной стратегии Пети, позволяющей ему выиграть своим четвертым ходом. Над пунктирной чертой указаны ходы игроков. Завершающие игру ходы подчеркнуты. Под пунктирной чертой в скобках приведены цепочки фишек после очередного хода.

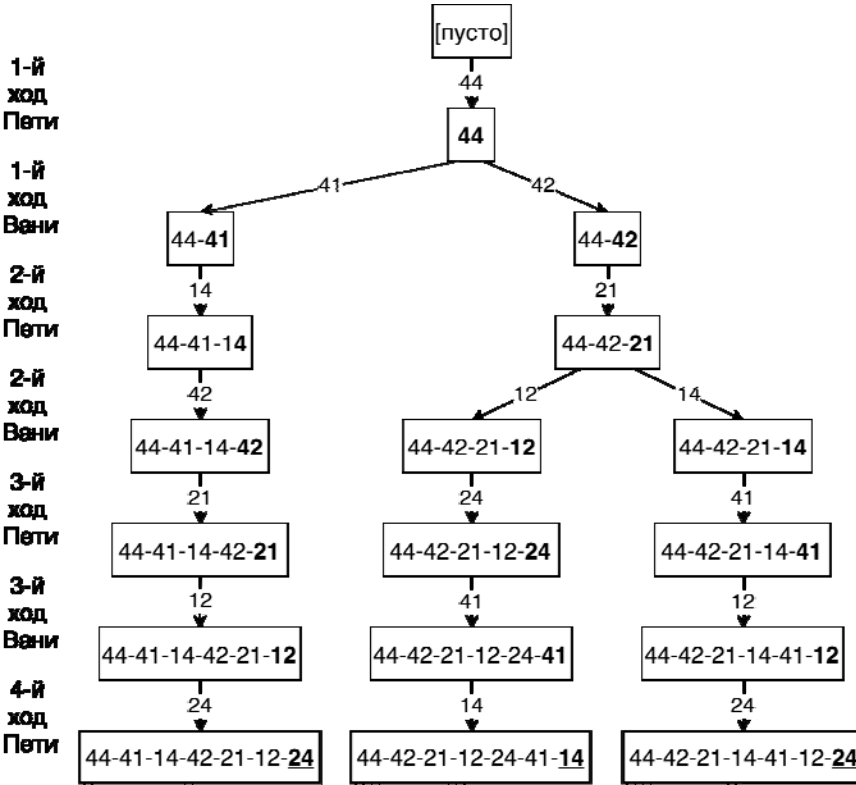


Рисунок 1. Дерево всех партий при выигрышной стратегии Пети, позволяющей ему выиграть своим четвертым ходом. В заключительных цепочках подчеркнут последний ход.

Примечание для проверяющего. И для таблицы, и для рисунка допускается сокращенная форма записи, когда указываются только ходы или только получающиеся цепочки фишек.

Задание 3.

Нужно убрать фишки 22 и 44.

Пример партии для сокращенного набора:

| Начальная позиция | 1-й ход Пети | 1-й ход Вани | 2-й ход Пети | 2-й ход Вани | 3-й ход Пети | 3-й ход Вани |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| [пусто] | 24 | 41 | 12 | <u>21</u> | 14 | <u>42</u> |

Примечание для проверяющего. Ученик может описать последовательность ходов в любой форме, главное, чтобы для приведенной им в качестве примера партии все ходы были указаны явно и в правильной последовательности.

Если убраны фишки 22 и 44, то игроками при любом развитии игры будут выставлены или 4 или 6 фишек. Поскольку 4 и 6 – чётные числа, то последнюю фишку всегда поставит второй игрок – Ваня. Возможны 24 различных партии в зависимости от первого хода Пети:

12-21-14-41

12-21-14-42-24-41

12-24-41-14-42-21

12-24-42-21-14-41

21-12-24-41-14-42

21-12-24-42

21-14-41-12-24-42

21-14-42-24-41-12

...

Остальные партии формируются аналогично. Ученик может не писать такого подробного комментария и привести пример только одной партии.

| Указания по оцениванию | Баллы |
|--|-------|
| Задание приравнивается к выполненному верно, если при правильно указанном количестве ходов не более чем в одной из партий , приведенных в ответе на задание, не более чем один из ходов указан неправильно, и при этом неверно указанный ход не является ни первым, ни последним . <i>Пример 1.</i> Пусть в ответе на задание 1 ученик вместо последовательности ходов 12-21-14-41 указал 12-21-41-42. Задание приравнивается к выполненному верно. <i>Пример 2.</i> Пусть в ответе на задание 1 ученик вместо последовательности ходов 12-21-14-41 указал 21-21-14-41. Задание считается выполненным неверно | |

| | |
|--|----------|
| Все три задания выполнены верно | 3 |
| Не выполнены условия, позволяющие поставить 3 балла, при этом верно выполнены любые два задания | 2 |
| Не выполнены условия, позволяющие поставить 2 балла, при этом верно выполнено одно любое задание | 1 |
| Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл | 0 |
| <i>Максимальный балл</i> | <i>3</i> |

27

На вход программы поступает последовательность из N целых положительных чисел, все числа в последовательности различны. Рассматриваются все пары различных элементов последовательности (элементы пары не обязаны стоять в последовательности рядом, порядок элементов в паре не важен). Необходимо определить количество пар, для которых произведение элементов делится на 26.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$). В каждой из последующих N 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 26.

Пример входных данных:

4

2

6

13

39

Пример выходных данных для приведённого выше примера входных данных:

4

Пояснение. Из четырёх заданных чисел можно составить 6 попарных произведений: 2·6, 2·13, 2·39, 6·13, 6·39, 13·39 (результаты: 12, 26, 78, 78, 234, 507). Из них на 26 делятся 4 произведения (2·13=26; 2·39=78; 6·13=78; 6·39=234).

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 Кбайт 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

(допускаются иные формулировки ответа, не искажающие его смысла)

Произведение двух чисел делится на 26, если выполнено одно из следующих условий (условия не могут выполняться одновременно).

А. Оба сомножителя делятся на 26.

Б. Один из сомножителей делится на 26, а другой не делится.

В. Ни один из сомножителей не делится на 26, но один сомножитель делится на 2, а другой – на 13.

Примечание для проверяющего. Условие делимости произведения на 26 можно сформулировать проще, например, так:

(один из сомножителей делится на 26) ИЛИ

(один сомножитель делится на 2, а другой – на 13).

Но в этом случае пара сомножителей может удовлетворять обоим условиям, что затруднит подсчёт количества пар.

При вводе чисел можно определять, делится ли каждое из них на 26, 2 и 13, и подсчитывать следующие значения:

1) n_{26} – количество чисел, кратных 26;

2) n_{13} – количество чисел, кратных 13, но не кратных 26;

3) n_2 – количество чисел, кратных 2, но не кратных 26.

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

Количество пар, удовлетворяющих условию А, можно вычислить по формуле $n_{26} \cdot (n_{26} - 1) / 2$.

Количество пар, удовлетворяющих условию Б, можно вычислить по формуле $n_{26} \cdot (N - n_{26})$.

Количество пар, удовлетворяющих условию В, можно вычислить по формуле $n_2 \cdot n_{13}$.

Поэтому искомое количество пар вычисляется по формуле

$$n_{26} \cdot (n_{26} - 1) / 2 + n_{26} \cdot (N - n_{26}) + n_2 \cdot n_{13}.$$

Ниже приведена реализующая описанный алгоритм программа на языке Паскаль (использована версия PascalABC)

Пример 1. Программа на языке Паскаль. Программа эффективна по времени и по памяти

```
var
  N: integer;           {количество чисел}
  a: integer;           {очередное число}
  n26, n13, n2: integer;
  k26: integer;         {количество требуемых пар}
  i: integer;

begin
  readln(N);
  n26:=0; n13:=0; n2:=0;
  for i:=1 to N do begin
    readln(a);
    if a mod 26 = 0 then
      n26 := n26+1
    else if a mod 13 = 0 then
      n13 := n13 + 1
    elseif a mod 2 = 0 then
      n2 := n2 + 1;
  end;
  k26 := n26*(n26-1) div 2 + n26*(N-n26) + n2*n13;
  writeln(k26)
end.
```

Комментарии для проверяющего

1. 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.

2. Общая идея решения, эффективного по времени, состоит в следующем. Просматриваем по очереди все элементы последовательности и накапливаем значения вспомогательных величин (в приведённом решении это счётчики n_2 , n_{13} , n_{26}). 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. Например, можно вместо n_2 и n_{13} использовать величины p_2 и p_{13} – количества чисел, которые делятся соответственно на 2 и на 13. Так как $n_2 = p_2 - n_{26}$ и $n_{13} = p_{13} - n_{26}$, то итоговая формула примет вид:

$$n_{26} \cdot (n_{26} - 1) / 2 + n_{26} \cdot (N - n_{26}) + (p_2 - n_{26}) \cdot (p_{13} - n_{26}).$$

Ещё один возможный вариант (есть и другие!) – подсчёт количества чисел, которые не делятся на 26, – можно вести по формуле $n_2 + n_{13} + nx$, где nx – количество чисел, которые не делятся ни на 2, ни на 13. Значение nx можно вычислить с помощью отдельного счётчика. Такая программа на языке Бейсик приведена ниже.

Все подобные программы оцениваются в 4 балла.

При любом наборе вспомогательных величин возможны различные способы записи итоговой формулы. Можно, например, раскрывать скобки и приводить подобные члены или, наоборот, выносить за скобки общие множители; можно вводить дополнительные переменные для отдельных слагаемых, а затем вычислять их сумму. Допустим любой способ записи вычислений, эквивалентный правильной формуле, выбранный способ записи не влияет на оценку.

3. 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.

4. 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. критерии)

Пример 2. Программа на языке Бейсик. Программа эффективна по времени и по памяти, но использует формулы, отличные от формул программы из примера 1

```
N26 = 0
N2 = 0
N13 = 0
NX = 0
INPUT N
FOR I = 1 TO N
  INPUT A
  IF A MOD 26 = 0 THEN
    N26 = N26 + 1
  ELSE
    IF A MOD 13 = 0 THEN
      N13 = N13 + 1
    ELSE
      IF A MOD 2 = 0 THEN
        N2 = N2 + 1
      ELSE NX = NX + 1
    END IF
  END IF
END IF
NEXT I
K26 = N26*(N26-1)\2 + N26*(N2+N13+NX) + N2*N13
PRINT K26
```

| Указания по оцениванию | Баллы |
|--|-------|
| <p>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 без программы не оценивается.</p> <p>Общий принцип оценивания можно неформально описать так.</p> <p>Эффективная правильная программа (возможно, с небольшим количеством синтаксических ошибок, подробнее см. ниже в критериях) оценивается 4 баллами.</p> <p>1 балл снимается за наличие одной содержательной ошибки (примерный список ошибок см. ниже в критериях).</p> <p>1 балл снимается за хранение исходных данных в массиве или другой аналогичной структуре, размер которой растёт с ростом количества элементов N</p> | |
| <p><i>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</i></p> <p><i>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</i></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, не допустимая для соответствующего типа данных. <p><i>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</i></p> | 4 |

| | |
|---|---|
| <p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введенных чисел; правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p> <p>Количество синтаксических ошибок («описок») указанных в критериях на 4 балла, не более пяти.</p> <p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> 1) допущена ошибка при вводе данных, например, не считывается значение N или числа могут быть считаны, только если будут записаны в одной строке через пробел; 2) ошибка при инициализации или отсутствие инициализации счётчиков; 3) в программе перепутаны знак целочисленного деления и взятия остатка или знаки операций «равно» и «не равно», «or» вместо «and» и т.п.; 4) использована неверная структура проверок, в результате которой некоторые счётчики могут получить неверное значение; 5) получены правильные значения счётчиков (вспомогательных величин), которые в принципе позволяют получить требуемое количество пар, но формула для вычисления записана неверно (комбинаторная ошибка); 6) отсутствует вывод ответа, или выводится значение не той переменной; 7) в описании алгоритма правильно описан смысл используемых вспомогательных величин, и в программе правильно записан алгоритм вычисления искомого количества пар, исходя из этих величин, однако при вычислении одной из вспомогательных величин допущена ошибка. <p>3 балла также ставится за программу, в которой нет содержательных ошибок, но используемая память зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в C++ или другой аналогичной структуре данных).</p> | 3 |
|---|---|

| | |
|--|---|
| <p>Не выполнены условия, позволяющие поставить 3 или 4 балла.</p> <p>Программа работает в целом верно, эффективно или нет.</p> <p>В реализации алгоритма допускается до трёх содержательных ошибок, описанных в критериях на 3 балла.</p> <p>Количество синтаксических ошибок, указанных в критериях на 4 балла, не должно быть более девяти.</p> <p>2 балла также ставится за корректное переборное решение, в котором все числа сохраняются в массиве (или другой аналогичной структуре), рассматриваются все возможные пары и подсчитывается количество подходящих произведений. Пример фрагмента соответствующей программы на языке Паскаль:</p> <pre> k := 0; for i := 1 to n - 1 do for j := i + 1 to n do if a[i]*a[j] mod 26 = 0 then k := k + 1; writeln(k); </pre> <p>В реализации переборного алгоритма не допускаются логические ошибки, например, когда учитываются произведения вида $a[i]*a[i]$ или пары считаются дважды.</p> | 2 |
| <p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа описывает в целом правильный алгоритм (эффективный или нет), но количество допущенных ошибок не укладывается в описанные выше ограничения.</p> | 1 |
| <p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла.</p> | 0 |
| Максимальный балл | 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 24–27, то третий эксперт проверяет ответы только на те задания, которые вызвали столь существенное расхождение.