

Ошибки при выполнении заданий второй части ЕГЭ по информатике (задания 24, 25, 26, 27)

Использованы материалы:

fipi.ru

inf-ege.sdamgia.ru

Пример задания 24 (вариант 1)

24

На обработку поступает натуральное число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран минимальную цифру числа, кратную 4. Если в числе нет цифр, кратных 4, 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.

Паскаль

```
var N,digit,minDigit: longint;
begin
  readln(N);
  minDigit := N mod 10;
  while N > 0 do
  begin
    digit := N mod 10;
    if digit mod 4 = 0 then
      if digit < minDigit then
        minDigit := digit;
    N := N div 10;
  end;
  if minDigit = 0 then
    writeln('NO')
  else
    writeln(minDigit)
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 123.
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.

Пример 1

N 24 1) Ответ: 3

2) Ответ: 114

3) а) Ошибочная строка: $\text{minDigit} := N \bmod 10$

Исправленная строка: $\text{minDigit} := 11$

б) Ошибочная строка: $\text{if minDigit} = 0 \text{ then}$

Исправленная строка: $\text{if minDigit} = 11 \text{ then.}$

Пример 2

N 24,

1. 3

2. 124

3. 1) $\text{if digit} < \text{minDigit} \text{ then}$
 ~~$\text{if digit} < \text{minDigit} \text{ or } \text{minDigit}$~~
 $\text{if } (\text{digit} < \text{minDigit}) \text{ or } (\text{minDigit} \bmod 4 \neq 0) \text{ then}$

2) $\text{if minDigit} = 0 \text{ then}$

$\text{if minDigit} \bmod 4 \neq 0 \text{ then}$

Пример задания 25 (ДемOVERсия 2018)

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.

Паскаль

```
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.
```

Пример задания 25 (Демоверсия 2018)

Критерии оценивания

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

```
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
```

Пример задания 25 (Демоверсия 2018)

Критерии оценивания

<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);11) используется переменная, не объявленная в разделе описания переменных;12) не указано или неверно указано условие завершения цикла;13) индексная переменная в цикле не меняется (например, в цикле <code>while</code>) или меняется неверно.	1
---	---



ФИПИ

Пример задания 26 (вариант 1)

26

Два игрока, Петя и Ваня, 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, ББ, БАОБАБ, БАР}.

Первым ходом Петя пишет Б (он мог написать Б или А).

Ваня 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.



ФИПИ

Пример задания 26

(продолжение)

Задание 1. а) Укажите, у кого есть выигрышная стратегия при исходном наборе слов {АБВГДАБВГДХ, ДГВБАДГВБА}. Опишите эту стратегию.

б) Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ТРИТРИ...ТРИ, РИТАРИТА...РИТА} (в первом слове ТРИ повторено 33 раза, т.е. его длина 99 букв; во втором слове РИТА повторено 44 раза, т.е. его длина 176 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 {СОЛНЦЕ, СОВА, СОВЕТ, ПРОСО, ПРОХОР, ПРОИЗВОДНАЯ}. У кого из игроков есть выигрышная стратегия для этого набора? Приведите в виде рисунка или таблицы дерево всех партий, возможных при этой стратегии.

Примечание для проверяющего. Можно поменять первую букву в слове ДГВБАДГВБА с буквой А в середине слова, это тоже приводит к верному решению.

Задание 3. Выигрышная стратегия есть у Вани. Дерево всех партий для этой стратегии показано в таблице 1 и на рисунке 1. *Примечание для проверяющего.* Оба способа изображения допустимы. Ученику достаточно привести один из них.

Начальная позиция	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)	3-й ход Пети (разобраны все ходы)	3-й ход Вани (только ход по стратегии)
[пусто]	С	СО	СОЛ	СОЛН	СОЛНЦ	<u>СОЛНЦЕ</u>
			СОВ	<u>СОВА</u>	-	
	П	ПР	ПРО	ПРОХ	ПРОХО	<u>ПРОХОР</u>

Таб. 1. Дерево всех партий при описанной стратегии Вани. Подчёркнуты позиции, в которых партии заканчиваются. Указаны только позиции, полученные после ходов игроков. Сам ход – это последняя буква полученного слова. *Примечание для проверяющего.* Ученик может не писать такого подробного комментария.

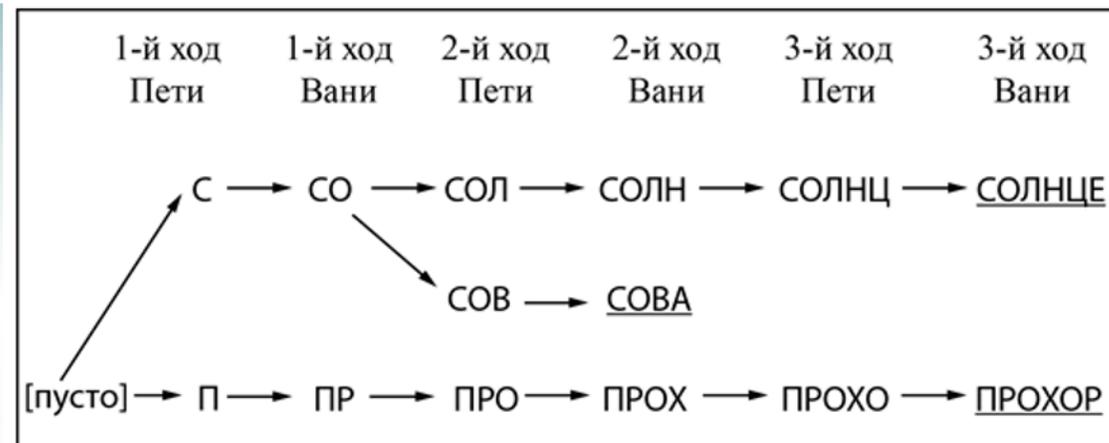


Рис. 1. Дерево всех партий при описанной стратегии Вани. Подчёркнуты позиции, в которых партии заканчиваются. Указаны только позиции, полученные после ходов игроков. Сам ход – это последняя буква полученного слова. *Примечание для проверяющего.* Ученик может не писать такого подробного комментария.



ФИПИ

Пример задания 27 (вариант 1)

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 14.

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 14.

Пример входных данных:

4
2
6
7
21

Пример выходных данных для приведённого выше примера входных данных:

4

Пояснение. Из четырёх заданных чисел можно составить 6 попарных произведений: $2 \cdot 6$, $2 \cdot 7$, $2 \cdot 21$, $6 \cdot 7$, $6 \cdot 21$, $7 \cdot 21$ (результаты: 12, 14, 42, 42, 126, 147).

Из них на 14 делятся 4 произведения ($2 \cdot 7 = 14$; $2 \cdot 21 = 42$; $6 \cdot 7 = 42$; $6 \cdot 21 = 126$).

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.

<p>Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает верно, эффективно по времени при условии исправления не более трёх содержательных ошибок, описанных в критериях на 3 балла, и не более девяти синтаксических ошибок, указанных в критериях на 4 балла. 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 14 = 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
<p><i>Максимальный балл</i></p>	4

Пример 3

```
Pascal ABC:  
const d = 1;  
var N, i, j, k: integer;  
    a: array[1..70000] of integer;  
Begin  
    readln(N);  
    for i := 1 to N do readln(a[i]);  
    k := 0; {argument return}  
    for i := 1 to N-d do begin  
        for j := i+d to N do begin  
            if (a[i] * a[j]) mod 14 = 0 then  
                k := k + 1;  
        end;  
    end;  
    writeln(k);  
End.
```

Пример 4

```
N = int(input())  
kr2 = 0  
kr7 = 0  
kr14 = 0  
for i in range(N):  
    a = int(input())  
    if a % 14 == 0:  
        kr14 += 1  
    elif elif a % 7 == 0:  
        kr7 += 1  
    elif a % 2 == 0:  
        kr2 += 1  
  
if kr2 != 0 and  
sum_p = kr2 * kr7 + kr14 * (N-1) kr14 * (N - kr14)  
print(sum_p)
```

if (a mod 14=0) then
inc(kr14)
if (a mod 14!=0) and (a mod 7=0)
inc(kr7)
if (a mod 14!=0) and (a mod 2=0)
inc(kr2)