

Объяснение задания 24 ЕГЭ

24-е задание характеризуется, как задание повышенного уровня сложности, время выполнения – примерно 30 минут

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ: ПОИСК ОШИБОК

Для решения 24 задания потребуется вспомнить следующие темы (за основу берется язык Паскаль — наиболее популярный язык программирования в Российских школах):

- Для организации ветвления используется условный оператор **if else**
- При выполнении нескольких операторов внутри условия, необходимо использовать так называемые **операторные скобки (составной оператор)**
- При необходимости использования нескольких условий в одном используются **логические операции**
- Цикл с постусловием **repeat**
- Цикл с предусловием **while**
- Цикл со счетчиком **for**

Разбор задачи ЕГЭ № 24

Операции целочисленного деления в Паскале.

Операции целочисленного деления `div`, `mod` предназначены только для работы целыми числами. Операция `div` определяет целую часть от деления, операция `mod` определяет остаток от деления.

Пример:

$$15 \text{ div } 2 = 7$$

$$15 \text{ mod } 2 = 1$$

Операции целочисленного деления `div`, `mod` играют большую роль при составлении программ.

Рассмотрим задачи, в которых используются `div`, `mod` [4]

Задача № 1

<http://www.e-olymp.com/ru/problems/4716>

Делёж яблок - 1

Условие:

n школьников делят k яблок поровну, неделящийся остаток остаётся в корзинке. Сколько яблок достанется каждому школьнику?

входные данные

Два положительных целых числа n и k, не превышающие 1500 - редко в школе бывает больше учеников, да и много яблок тоже кушать вредно...

Выходные данные

Вывести количество яблок, которое достанется каждому школьнику.

```
program pr1;  
var r,k,n:integer;  
begin  
  readln(n,k);  
  r:=k div n;  
  writeln(r);  
end.
```

Задача № 2

Делёж яблок - 2

Условие:

n школьников делят k яблок поровну, неделящийся остаток остаётся в корзинке. Сколько яблок останется в корзинке?

Входные данные

Два положительных целых числа n и k, не больших 1500 - редко в школе бывает больше учеников, да и где найти такую корзинку?

Выходные данные

Вывести количество яблок, которое останется в корзинке.

```
program pr1;  
var r,k,n:integer;  
begin  
  readln(n,k);  
  r:=k mod n;  
  writeln(r);  
end.
```

Задача № 3. Найти сумму цифр натурального числа N, где $N < 10^9$

```
Program ex1;
```

```
Var N, digit, sum: longint;
```

```
begin
```

```
  readln(N);
```

```

sum := 0;

while N > 0 do

begin

    digit := N mod 10;

    sum := sum+digit;

    N := N div 10;

end;

writeln(sum)

end.

```

Составим трассировочную таблицу для N=357

№	условие	digit	sum	N
1	357>0	357 mod 10= 7	0+7=7	357 div 10 =35
2	35>0	35 mod 10=5	7+5=12	35 div 10 =3
3	3>0	3 mod 10=3	12+3=15	3 div 10 = 0
4	0>0			

Решение 24 заданий ЕГЭ по информатике

Разбор 24 задания ЕГЭ по информатике 2017 года (один из вариантов со слов выпускника):

Ученик написал программу, которая находит максимальную цифру заданного числа, кратную 6 (примечание: 0 кратен любому числу). Если таких чисел нет, то программа должна напечатать NO. Но его программа оказалась неверной.

```

1
2
3 var N,digit,maxDigit:integer;
4 begin
5   readln(N);
6   maxDigit := N mod 10;
7   while N>0 do begin
8     digit:= N mod 10;
9     if (digit > maxDigit) and (digit mod 6 = 0) then
10      maxDigit:=digit;
11     N:=N div 10;
12   end;
13 if maxDigit = 0 then writeln('NO')
14 else writeln(maxDigit);
15 end.
1
3

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 143.
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). Для каждой ошибки выпишите строку, в которой сделана ошибка и укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Решение:

Рассмотрим алгоритм:

- Алгоритм предназначен для поиска максимальной цифры числа, кратной 6.
- Переменная **maxDigit** предусмотрена для хранения максимальной цифры числа. Изначально ей в программе присвоена крайняя справа цифра (разряд единиц) — `maxDigit := N mod 10;`. Т.е. остаток при делении числа на 10 — это и есть крайняя справа цифра числа.
- В цикле пока `N > 0`:
 - от числа **N** «отрезается» каждая цифра справа и сохраняется в переменную **digit**.
 - осуществляется проверка: если «отрезанная» цифра больше максимальной (`digit > maxDigit`) и кратна **6** (`digit mod 6 = 0`), то устанавливаем ее максимальной (`maxDigit:=digit;`)
 - сохраняем **N** без данной «отрезанной» цифры (`N:=N div 10;`, т.е. целочисленно делим на **10**, таким образом отсекая цифру справа)
 - После цикла осуществляется проверка: если максимальным числом был **0**, то выводим слово *NO*, иначе выводим максимальное число.

Решение заданий:

1. При вводе числа **143** на экран выведется цифра **3**

Дальнейшее объяснение НЕ нужно писать на экзамене:

- В четвертой строке программы в **maxDigit** сохранится крайняя правая цифра числа, т.е. **3**.
- В цикле в условном операторе (строка 7) больше не найдется ни одной цифры, удовлетворяющей условию, т.е. ни **4**, ни **1** не будут кратными **6**.
- В результате программа выведет значение **maxDigit**, т.е. цифру **3**.

Рассмотрим подробно каждый шаг:

```
maxDigit := 3;
```

```

1 шаг:
while 143>0 do begin
    digit:= 3;
    if (3 > 3)and (3 mod 6 = 0) then условие ложно
        maxDigit:=digit; - не выполняется
    N:=14;
end;

2 шаг:
while 14>0 do begin
    digit:= 4;
    if (4 > 3)and (4 mod 6 = 0) then условие ложно
        maxDigit:=digit; - не выполняется
    N:=1;
end;

3 шаг:
while 1>0 do begin
    digit:= 1;
    if (1 > 3)and (1 mod 6 = 0) then условие ложно
        maxDigit:=digit; - не выполняется
    N:=0;
end;

4 шаг:
while 0>0 do begin условие ложно
    ...
end;
if maxDigit = 0 then условие ложно
    writeln('NO') - не выполняется
else writeln(maxDigit); - вывод числа 3

```

2. При вводе любого числа, крайней справа цифрой которого является цифра **6**, а остальные цифры будут меньше либо равны ей, программа выдаст правильный результат. Например, ввод числа **126** вернет верный результат.

Дальнейшее объяснение НЕ нужно писать на экзамене:

в **maxDigit** сохранится цифра **6**, которая максимальна в числе и кратна **6**.

3.

1) неверная инициализация переменной maxDigit:

ошибочная строка: `maxDigit := N mod 10;` (№ 4)

исправление: `maxDigit := -1;`

Дальнейшее объяснение НЕ нужно писать на экзамене:

Изначально максимальным необходимо назначить наименьшую возможную цифру, которая не кратна 6 (0 кратен любому числу, поэтому -1)

2) неверное условие:

ошибочная строка: `if maxDigit = 0 then` (№ 11)

исправление: `if maxDigit = -1 then`

Дальнейшее объяснение НЕ нужно писать на экзамене:

Вытекает из предыдущего пункта

Подробное **решение 24 задания** посмотрите на видео:

Разбор 24 задания ЕГЭ по информатике 2017 года ФИПИ вариант 5 (Крылов С.С., Чуркина Т.Е.):

Дано натуральное число N , не превосходящее **1000**. Необходимо определить, существует ли такое натуральное число M , что N можно представить в виде произведения $(M-1)(M+1)$. Если это возможно, то следует напечатать число M , в противном случае вывести сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

```
1
2 var n, m: integer;
3 begin
4   read(n);
5   m:=2;
6   while ((m-1)*(m+1) = n) and (m<n) do
7     m:=m+1;
8   if m < n then
9     writeln(n)
10  else
11    writeln('Не существует')
12 end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа **3**.
2. Приведите пример натурального числа, при вводе которого приведенная программа напечатает то, что требуется.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

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

Решение:

1. Программа выведет **'Не существует'**

Дальнейшее объяснение НЕ нужно писать на экзамене:

```
n = 3;
m = 2;

1 шаг:
while ((2-1)*(2+1) = 3) and (2<3) do условие истинно
    m:=2+1;

2 шаг:
while ((3-1)*(3+1) = 3) and (2<3) do условие ложно
    ...
if 3 < 3 then условие ложно
    writeln(n) не выполняется
else
    writeln('Не существует') выполняется
end.
```

2. При вводе числа **2** (напечатает **'Не существует'**)

3.

1) ошибочное условие цикла:

ошибочная строка: `while ((m-1)*(m+1) = n)`

исправление: `while ((m-1)*(m+1) <> n)`

2) ошибочный вывод:

ошибочная строка: `writeln(n)`

исправление: `writeln(m)`