

ИТОГОВЫЙ КОНСПЕКТ

Тема №24

ОШИБКИ

1. ИНИЦИАЛИЗАЦИЯ ПЕРЕМЕННЫХ

МАКСИМУМ ИЗНАЧАЛЬНО ДОЛЖЕН БЫТЬ
МИНИМАЛЕН

$\max = -10000$ (например)

МИНИМАЛЕН ИЗНАЧАЛЬНО ДОЛЖЕН БЫТЬ
МАКСИМУМ

$\min = 10000$ (например)

СУММА ИЗНАЧАЛЬНО ДОЛЖНА БЫТЬ РАВНА 0

ПРОИЗВЕДЕНИЕ ИЗНАЧАЛЬНО ДОЛЖНО БЫТЬ
РАВНО 1

2. ПОДСЧЕТ СУММЫ ЦИФР/КОЛИЧЕСТВА

К ПРЕДЫДУЩЕМУ ЗНАЧЕНИЮ ДОЛЖНА ПРИБАВЛЯТЬСЯ
НОВАЯ ЦИФРА

$\text{sum} = \text{sum} + x$

ПРИ ПОДСЧЕТЕ КОЛИЧЕСТВА

К ПРЕДЫДУЩЕМУ ЗНАЧЕНИЮ ДОЛЖНА ПРИБАВЛЯТЬСЯ
ЕДИНИЦА

$\text{count} = \text{count} + 1$

3. ВЫВОД

ЧАСТО ВЫВОДИТСЯ НЕ ТО, ЧТО ДОЛЖНО БЫТЬ В
УСЛОВИИ

4. УСЛОВНЫЕ ОПЕРАТОРЫ

ЧАСТО ПРИ ПОИСКЕ МАКС/МИН НЕПРАВИЛЬНЫЙ ЗНАК,
ТАКЖЕ БЫВАЮТ И ДРУГИЕ МОМЕНТЫ, ПОЭТОМУ
ВНИМАТЕЛЬНЕЕ КО ВСЕМ УСЛОВНЫМ ОПЕРАТОРАМ

5. СТРОГОСТЬ ЗНАКОВ

ЧАСТО В УСЛОВИИ И В ПРОГРАММЕ ЗНАКИ ДОЛЖНЫ
БЫТЬ ПРОТИВОПОЛОЖНЫМИ!



ИТОГОВЫЙ КОНСПЕКТ

Тема №24

ПРИМЕР ЗАДАНИЯ

Дано целое положительное число A . Требуется вывести такое минимально возможное нечётное натуральное число K , при котором сумма квадратов первых нечётных чисел $1^2 + 3^2 + \dots + K^2$ окажется больше A . Для решения этой задачи ученик написал программу, но, к сожалению, его программа – неправильная.

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 11.
2. Укажите наименьшее значение A , при котором программа выведет верный ответ.
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.

1.

А) Итак, давайте прогоним число 11 через программу $s = 0$ $k = 1$ $a = 11$

Б) Проверяем условие $s \leq A$ верно, Идем дальше

В) $s := s + k * k$; $s = 0 + 1 = 1$
 $k := k + 1$; $k = 1 + 1 = 2$

Д) Повторяем, начиная с шага Б

$1 \leq 11$ да
 $s := s + k * k$; $s = 1 + 4 = 5$
 $k := k + 1$; $k = 2 + 1 = 3$

$5 \leq 11$ да
 $s := s + k * k$; $s = 5 + 9 = 14$
 $k := k + 1$; $k = 3 + 1 = 4$

$14 \leq 11$ нет
вышли из цикла
печатаем k , ТО ЕСТЬ 4
ОТВЕТ : 4

```
var a, s, k: integer;
begin
  read(A);
  s := 0;
  k := 1;
  while s <= A do begin
    s := s + k*k;
    k := k + 1;
  end;
  writeln(k)
end.
```

2.

А) Минимальное значение A , при котором программа работает правильно, равно 1. Так как мы один раз зайдём в цикл, а также так как это пограничное значение!
ОТВЕТ: 1

3.

А) Первое, что бросается в глаза $k := k + 1$; – это неверно (получается идем по всем k , хотя должны только по нечетным!), давайте заменим

На
 $k := k + 2$;

Б) Неверный вывод! $writeln(k)$
Так как мы k увеличиваем после того как меняем s , ответ у нас выводится больше чем нужно! Поэтому стоит поменять на $writeln(k-2)$



ИТОГОВЫЙ КОНСПЕКТ

Тема №24

ПРИМЕР ЗАДАНИЯ

Дано целое положительное число N , не превосходящее 1000. Необходимо определить, является ли это число степенью числа 5. То есть требуется определить, существует ли такое целое число K , что $5^K = N$, и вывести это число либо сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

Pascal	Python
<pre>var n, k: integer; begin read(n); k := 0; while k mod 5 = 0 do begin k := k + 1; n := n div 5; end; if k = 1 then writeln(k) else writeln('Не существует') end.</pre>	<pre>n = int(input()) k = 0 while k % 5 == 0: k = k + 1 n = n // 5 if k == 1: print(k) else: print('Не существует')</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 25.
2. Приведите пример числа, при вводе которого приведённая 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.

1) Для того чтобы выполнить первое задание, используем ручную прокрутку:

Поскольку после завершения цикла $k=1$, то **для входного числа $n=25$ будет выведено значение 1.**

2) Для ответа на второй вопрос обратим внимание, что конечное значение переменной k никак не зависит от n , и программа ВСЕГДА будет выводить результат 1. Поэтому **программа работает правильно только при $n=5^1=5$.**

3) Чтобы исправить программу, нужно понять алгоритм её работы, как задумывал автор (но не реализовал). Очевидно, что k – это счётчик, с помощью которого мы считаем степень числа 5. Оператор **$n := n \text{ div } 5$;**

говорит о том, что предполагалось делить исходное число на 5 до тех пор, пока оно делится, и считать, сколько раз нам удалось разделить его на 5 без остатка. Поэтому условие работы цикла должно быть записано как «пока n делится на 5», то есть

while $n \bmod 5 = 0$ do begin

...

end;

Одна ошибка найдена.

4) Если введённое число – это степень числа 5, то оно будет делиться на 5 до тех пор, пока в переменной n не останется 1. Таким образом, условие «число представляет собой натуральную степень числа 5» после цикла запишется в виде

if $n = 1$ then

...

Это вторая ошибка.

5) Итак, ответ на третий вопрос таков:

в программе нужно исправить две ошибки

Неверное условие цикла:

Было: $\text{while } k \bmod 5 = 0 \text{ do begin}$

Исправление: $\text{while } n \bmod 5 = 0 \text{ do begin}$

Неверное условие в условном операторе:

Было: $\text{if } k = 1 \text{ then}$

Исправление: $\text{if } n = 1 \text{ then}$

