

Исправление ошибок в простой программе с условными операторами

ЕГЭ

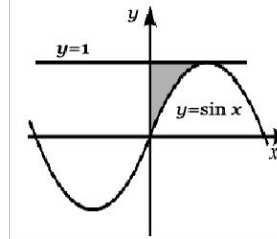
24

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x , y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно.

Вот она:

```
var x,y: real;
begin
  readln(x,y);
  if y <= 1 then
    if x >= 0 then
      if y >= sin(x) then
        write(' принадлежит')
      else write(' не принадлежит')
    end.
end.
```

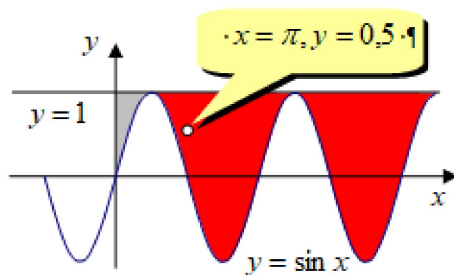
Последовательно выполните следующее: 1) Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).



```
СИ++
#include <iostream>
#include <cmath>
using namespace std;
```

```
int main()
{
  setlocale(0, "");
  double x, y;
  cin >> x >> y;
  if (y <= 1)
  if (x >= 0)
  if (y >= sin(x))
    cout << "принадлежит";
  else
    cout << "не принадлежит";
  return 0;
}
```

- 1) отметим на рисунке область, где выполняются все нужные условия, кроме $x \leq \frac{\pi}{2}$ (красная зона);
- 2) для всех точек, которые находятся в «красной» зоне программа выдаст сообщение «принадлежит», хотя в самом деле эти точки не принадлежат заданной области; одна из таких точек имеет координаты $x = \pi, y = 0,5$



функция достигает максимума при $x = \frac{\pi}{2}$, поэтому получаем условие $x \leq \frac{\pi}{2}$

2)

```
var x,y: real;
```

```
begin
```

```
  readln(x,y);
```

```
  if (x >= 0) and (x <= pi/2) and
```

```
    (y <= 1) and (y >= sin(x)) then
```

```
      write('принадлежит')
```

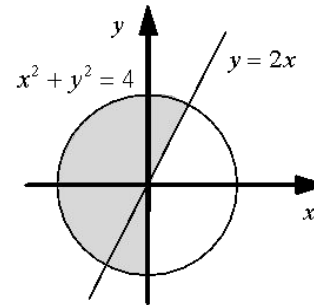
```
  else write('не принадлежит');
```

```
end.
```

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x , y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(0, "");
    double x, y;
    cin >> x >> y;
    if (y >= 2 * x)
    if (x <= 0)
    if (x * x + y * y <= 4)
    cout << "принадлежит";
    else
    cout << "не принадлежит";
    return 0;
}
```



Pascal

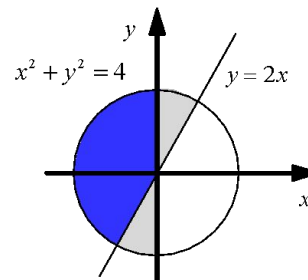
```
var x, y: real;
begin
    readln(x, y);
    if y >= 2*x then
    if x <= 0 then
    if x*x+y*y <= 4 then
    write('принадлежит')
    else
    write('не принадлежит')
    end.
end.
```

Последовательно выполните следующее: 1) Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

1) Во-первых, условия

```
y >= 2*x
x <= 0
x*x+y*y <= 4
```

выполненные одновременно, выделяют только область, обозначенную на рисунке синим цветом, а не всю заштрихованную область (серую). То есть для точек, попавших в серую область, но не попавших в синюю (например, для точки $(0,25; 0,55)$), ответ будет неверный.



Pascal

```
var x, y: real;
begin
    readln(x, y);
    if ((y >= 2*x) or (x <= 0)) and (x*x+y*y <= 4)
    then
    write('принадлежит')
    else
    write('не принадлежит');
    end.
```

Требовалось написать программу, которая вводит с клавиатуры натуральное число N , не превышающее 10^9 , и выводит сумму чётных цифр в десятичной записи этого числа или 0, если чётных цифр нет. Программист торопился и написал программу неправильно. Вот она:

```
var N: longint;  
    s: integer;  
begin  
  readln(N);  
  s := 0;                1981  
  while N > 1 do begin  
    if N mod 2 = 0 then begin  
      s := N mod 10;  
    end;  
    N := N div 10;  
  end;  
  write(s);  
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1984.
2. Приведите пример числа, при вводе которого программа выдаст 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.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество нечётных чисел в исходной последовательности и максимальное нечётное число. Если нечётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Вот она:

```
const n = 4;
var i, x: integer;
var maximum, count: integer;
begin
  count := 0;
  maximum := 999;           maximum := 0;
  for i := 1 to n do begin
    read(x);
    if x mod 2 <> 0 then begin
      count := count + 1;
      if x > maximum then maximum := 1           maximum := x
    end
  end;
  if count > 0 then begin
    writeln(count);
    writeln(maximum)
  end
  else writeln('NO')
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности: 2 9 4 3 2 999
2. Приведите пример такой последовательности, содержащей хотя бы одно нечётное число, что, несмотря на ошибки, программа печатает правильный ответ. 1 2 4 999
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. 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.

Требовалось написать программу, которая получает на вход натуральное число N, не превосходящее 10^9 , и выводит число, которое получается из N после удаления всех единиц; порядок остальных цифр при этом не меняется. Например, число 19520125 должно быть преобразовано в число 952025. Число, в котором все цифры – единицы и нули, должно быть преобразовано в 0. Незначащие нули в старших разрядах полученного числа печатать не нужно. Программист торопился и написал программу неправильно.

Pascal

```
var N, R, T: longint;
    d: integer;
begin
  readln(N);
  R:=0;
  T:=1;
  while N>0 do begin
    d := N mod 10;
    if d<>1 then begin
      R := R + d*T;
      T := T+1
    end;
    N := N div 10;
  end;
  writeln(T);
end.
```

Python

```
N = int(input())
R = 0
T = 1
while N > 0:
    d = N % 10
    if d != 1:
        R = R + d*T
        T = T + 1
    N = N // 10
print(T)
```

```
#include <iostream>
using namespace std;
```

```
int main()
{
    setlocale(0, "");
    long N, R = 0, T = 1;
    int d;
    cin >> N;
    while (N > 0)
    {
        d = N % 10;
        if (d != 1)
        {
            R += d*T;
            T ++;
        }
        N /= 10;
    }
    cout << T;
    return 0;
}
```

1) при вводе числа 314 программа выводит 3 (неправильный ответ).

2) *Ошибка 1:* выводить следует значение переменной R:

строка: `writeln(T);`

исправление: `writeln(R);`

Ошибка 2: изменение переменной T в теле цикла

строка: `T := T+1`

исправление: `T := T*10`

234

d=2

R=4+3*10=34

Дано целое положительное число А. Требуется вывести такое минимально возможное натуральное число К, при котором

$$1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+K)$$

окажется больше А. Для решения этой задачи ученик написал программу, но, к сожалению, его программа – неправильная.

| Pascal | Python |
|--|--|
| <pre>var A, s, p, k: integer; begin read(A); s := 0; p := 0; k := 1; while s < A do begin p := p + k; s := s + p; k := k + 1; end; writeln(k); end.</pre> | <pre>A = int(input()) s = 0 p = 0 k = 1 while s < A: p = p + k s = s + p k = k + 1 print(k)</pre> |

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 12.
2. Укажите любое значение А, при котором программа выведет верный ответ. Запишите этот 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.

СИ++

```
#include <iostream>
using namespace std;
```

```
int main()
{
  int A, s = 0, p = 0, k = 1;
  cin >> A;
  while (s < A)
  {
    p += k;
    s += p;
    k++;
  }
  cout << k;
  return 0;
}
```


С помощью ручной трассировки убеждаемся, что при вводе числа 12 программа выдаёт ответ 5.

| | p | s | k | A |
|-------------|----|----|---|----|
| read(A) | | | | 12 |
| s := 0 | | 0 | | |
| p := 0 | 0 | | | |
| k := 1 | | | 1 | |
| s < A → да | | | | |
| p := p + k | 1 | | | |
| s := s + p | | 1 | | |
| k := k + 1 | | | 2 | |
| s < A → да | | | | |
| p := p + k | 3 | | | |
| s := s + p | | 4 | | |
| k := k + 1 | | | 3 | |
| s < A → да | | | | |
| p := p + k | 6 | | | |
| s := s + p | | 10 | | |
| k := k + 1 | | | 4 | |
| s < A → да | | | | |
| p := p + k | 10 | | | |
| s := s + p | | 20 | | |
| k := k + 1 | | | 5 | |
| s < A → нет | | | | |

1) при вводе числа 12 будет выведено значение 5.

Замечаем, про цикл заканчивается, когда $s \geq k$, а нам нужно строгое неравенство. Поэтому необходимо поменять условие цикла на $s \leq a$. Кроме того, программа выводит следующее значение k, потому что k увеличивается после того, как увеличились переменные p и s. Поэтому нужно выводить значение k-1, меньшее на единицу.

3) в программе нужно исправить две ошибки

a. Неверное условие цикла:

Было: `while s < A do begin`

Исправление: `while s <= A do begin`

b. Неверный вывод результата:

Было: `writeln(k);`

Исправление: `writeln(k-1); {вернуться на шаг назад}`

Остается ответить на вопрос 2 – найти значение A, при котором программа выдаёт правильный ответ. Во-первых, подбором сразу находим, что при $A = 1$ программа выдаёт правильный ответ 2. Кроме того, есть и другие варианты. Из-за неверного условия цикл останавливается на шаг раньше, если $s = A$, при этом выводится следующее, то есть правильное значение k. Такая ситуация происходит при $A = 1$, $A = 1 + (1+2) = 4$, $A = 1 + (1+2) + (1+2+3) = 10$ и т.д.

2) при $A = 1$ программа выводит правильный ответ 2.

Путешественник пришел в 06:00 на автостанцию поселка Иваново и увидел следующее расписание автобусов:

| Станция отправления | Станция прибытия | Время отправления | Время прибытия | Стоимость, у.е. |
|---------------------|------------------|-------------------|----------------|-----------------|
| Иваново | Петрово | 6:00 | 7:00 | 15 |
| Петрово | Рамово | 8:00 | 9:30 | 20 |
| Петрово | Советово | 7:20 | 8:00 | 25 |
| Петрово | Ракитино | 6:45 | 8:30 | 35 |
| Рамово | Роцино | 10:00 | 11:00 | 30 |
| Петрово | Иваново | 12:00 | 13:00 | 25 |
| Советово | Роцино | 7:30 | 9:00 | 20 |
| Рамово | Петрово | 8:20 | 9:00 | 10 |
| Ракитино | Роцино | 9:00 | 10:00 | 15 |
| Роцино | Советово | 10:20 | 11:20 | 20 |

По таблице определите стоимость маршрута, выбрав который, путешественник раньше всего окажется в поселке Роцино.

Для удобства представим все интересующие нас маршруты в виде графа:



Определим маршрут, по которому путешественник раньше всего попадет в поселок Рощино:

Первый маршрут: Иваново-Петрово (6:00-7:00) → Петрово-Ракитино (6:45-8:30) → Ракитино-Рощино (9:00-10:00). Данный маршрут не подходит, т.к. в поселке Петрово путешественник окажется только в 7:00, а отправление оттуда в 6:45;

Второй маршрут: Иваново-Петрово (6:00-7:00) → Петрово-Советово (7:20-8:00) → Советово-Рощино (7:30-9:00). Маршрут не подходит, т.к. в поселке Советово путешественник окажется только в 8:00, а отправление оттуда в 7:30;

Третий маршрут: Иваново-Петрово (6:00-7:00) → Петрово-Рамово (8:00-9:30) → Рамово-Рощино (10:00-11:00).

Самый быстрый и единственный возможный маршрут – *третий*. Количество у.е., затраченных на поездку по данному маршруту: $15 + 20 + 30 = 65$ у.е.

Ответ: 65

Найдите наименьшее основание системы счисления, в которой десятичное число 91 имеет ровно три значащих разряда.

Если запись числа имеет ровно три значащих разряда, то запись числа трехзначна (состоит из трех цифр, не равных нулю). Если запись числа трехзначна, то максимальное значение числа равно $x^3 - 1$, где переменная - основание системы счисления. Это можно увидеть на примере с десятичной системой счисления. Максимальное четырехзначное число: $10^4 - 1 = 10000 - 1 = 9999$, максимальное двузначное число: $10^2 - 1 = 100 - 1 = 99$. Аналогично перебираем другие системы счисления, удовлетворяющие условию задачи:

Двоичная: $2^3 - 1 = 7$, слишком мало, запись числа 91 будет состоять более, чем из 3-х цифр.

Троичная: $3^3 - 1 = 26$, слишком мало, запись числа 91 будет состоять более, чем из 3-х цифр.

Четверичная: $4^3 - 1 = 63$, слишком мало, запись числа 91 будет состоять более, чем из 3-х цифр.

Пятеричная: $5^3 - 1 = 124$. Значит, искомое основание - 5. Для проверки переведем $91_{10} = 3 \cdot 5^2 + 3 \cdot 5^1 + 1 \cdot 5^0 = 331_5$

Ответ: 5