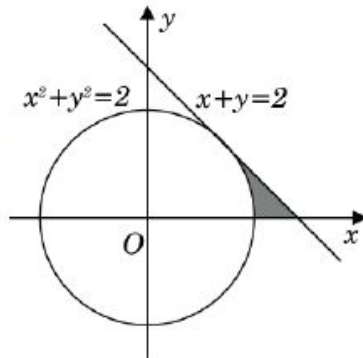


# Подготовка к ЕГЭ по информатике

## Поиск ошибок в программах (С1)

Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  – действительные числа) и определяется принадлежность этой точки заданной заштрихованной области (включая границы). Область ограничена осью абсцисс, окружностью  $x^2 + y^2 = 2$  и прямой  $x + y = 2$ . Заданные прямая и окружность касаются в точке  $x = 1, y = 1$ . Программист торопился и написал программу неправильно.



ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if x*x+y*y&gt;=2 then If y&gt;=0 then if x+y&lt;=2 then write ('принадлежит') else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF x*x+y*y&gt;=2 THEN IF y&gt;=0 THEN IF x+y&lt;=2 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f", &amp;x, &amp;y); if (x*x+y*y&gt;=2) if (y&gt;=0) if (x+y&lt;=2) printf ("принадлежит"); else printf("не принадлежит"); }</pre>

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, достаточно указать любой способ доработки исходной программы).

**C1- 1.** Требовалось написать программу, которая решает уравнение

$$ax + b = 0$$

относительно  $x$  для любых чисел  $a$  и  $b$ , введенных с клавиатуры. Все числа считаются действительными.

Программист торопился и написал программу неправильно.

**Program urav;**

**var**

**a, b, x: real;**

**begin**

**readln(a,b,x);**

**If b = 0 then**

**write('x = 0')**

**else**

**If a = 0 then**

**write('нет решений')**

**else**

**write('x =', -b/a);**

**end.**

## ***Последовательно выполните задания:***

- 1) Приведите пример таких чисел **a, b, x**, при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.
- 3) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы.

**Решение**

1. **a=0, b=0, x=0** (выдается 0, а должно быть любое число)
2. Лишняя часть программы – использование и ввод переменной **x**.

## Доработка программы:

```
Program C1_1;
```

```
var a,b: real;
```

```
begin
```

```
write('Введите a и b');
```

```
readln(a,b);
```

```
if a = 0 then
```

```
    if b = 0 then write('любое число')
```

```
        else write('нет решений')
```

```
    else
```

```
        write('x=', -b/a:8:2);
```

```
Readln;
```

```
End.
```

**C1-2.** В программе вводятся с клавиатуры три различных целых числа **a**, **b**, **c**. Далее требуется преобразовать данные так, чтобы в переменной **a** оказалось наименьшее число, в переменной **b** среднее число, а в переменной **c** - наибольшее.

Программист торопился и написал программу неправильно.

```
Program tri_chisla;  
Var a, b, c, x : integer;  
Begin  
    Write('Введите a, b, c');  
    readln(a, b, c);  
    If a>b then begin  
        x:= a; a:= b; b:= x;  
        end;  
    If b>c then begin  
        x:= b; b:= c; c:= x;  
        end;  
end.
```

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел **a, b, c**, после обработки которых исходной программой соотношение  $a < b < c$  не будет выполнено.
- 2) Исправьте ошибку в программе так, чтобы она полностью соответствовала постановке задачи.

**Решение**



**1) A=5, b=7, c=2** (Программа работает неправильно, если минимальное число вводится последним и попадает в переменную c).

## Доработка программы:

### **Program C1\_2;**

```
var a, b, c, x: integer;
```

```
Begin
```

```
  write('Введите три числа');  read(a, b, c);
```

```
  if a > b then begin x:=a; a:=b; b:=x; end;
```

```
  if b > c then begin x:=b; b:=c; c:=x; end;
```

```
  if a > b then begin x:=a; a:=b; b:=x; end;
```

```
  write('Упорядоченная последовательность чисел:');
```

```
  writeln(a:5, b:5, c:5);
```

```
  readln;
```

```
end.
```

**C1-3.** Требовалось написать программу, которая решает уравнение  $a|x|=b$  относительно  $x$  для любых чисел  $a$  и  $b$ , введенных с клавиатуры. Все числа считаются действительными.

Программист торопился и написал программу с ошибками:

```
Program modul;  
Var a, b, x : Real;  
Begin  
  Readln(a, b, x);  
  If a=0    then  
    If b=0 then Write('Любое число')  
      else Write('Решений нет')  
        else  
          If b=0 then Write('x=0')  
            else  
              Write('x=',b/a, 'или x=',-b/a);  
End.
```

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел **a, b, x**, при которых программа работает неправильно.
- 2) Укажите, какая часть программы является лишней.
- 3) Доработайте программу так, чтобы не было случаев ее неправильной работы . Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.

**Решение**

1.  $a=1, b=-1, x=0$  ( $a$  и  $b$  не могут иметь разные знаки)
2. Лишняя часть программы – использование и ввод переменной  $x$ .

### **Program C1\_3;**

```
var a,b: real;
```

```
begin
```

```
  readln(a, b);
```

```
  if a = 0 then
```

```
    if b = 0 then write ('x - любое число')
```

```
      else write ('нет решений')
```

```
    else
```

```
      if b = 0 then write('x = 0')
```

```
        else
```

```
          if a*b < 0 then write('нет решений')
```

```
            else
```

```
              write('x =',b/a:8:2,' или x =',-b/a:8:2);
```

```
Readln; end.
```

**Доработка программы:**

**C1-4.** Требовалось написать программу, которая решает неравенство

$$ax + b > 0$$

относительно **x** для любых чисел **a** и **b**, введенных с клавиатуры. Все числа считаются действительными.

Программист торопился и неправильно написал программу.

```
Program nerav1;  
Var a, b, x : real;  
Begin  
    readln(a,b,x);  
    if a=0 then  
        write( 'любое число')  
    else  
        if a>0 then  
            write( 'x > ',-b/a)  
        else  
            write( 'x < ',-b/a);  
End.
```

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел **a**, **b**, **x**, при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.
- 3) 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).

**Решение**



- 1)  $a=0$ ,  $b=-2$ ,  $x=1$  (значение  $x$  может быть не указано)
- 2) Лишняя часть программы – использование и ввод переменной  $x$ .

## Доработка программы:

### **Program C1\_4;**

```
Var a, b : real;
```

```
Begin
```

```
  readln(a, b);
```

```
  if a=0 then
```

```
    if b>0 then write( 'любое число')
```

```
      else write( 'нет решений')
```

```
    else
```

```
  if a>0 then write( 'x >',-b/a:8:2)
```

```
    else write( 'x <',-b/a:8:2);
```

```
  Readln;
```

```
End.
```

**C1- 5.** Требовалось написать программу, которая определяет, лежит ли точка  $A(x,y)$  внутри некоторого **кольца** («внутри» понимается в строгом смысле). Центр кольца находится в начале координат. Для кольца заданы внутренний и внешний радиусы  $R1$  и  $R2$ . Известно, что  $R1 \neq R2$ , но неизвестно какой из них больше. В том случае, когда точка  $A$  лежит внутри кольца, программа должна выводить соответствующее сообщение, в противном случае никакой выходной информации не выдается. Программист сделал в программе ошибки.

**Program kolco;**

**Var**

**x, y, r1,r2 : real;**

**h: real; {расстояние от т.А до  
начала координат}**

**Begin**

**Readln(x,y,r1,r2);**

**h:=sgrt(x\*x+y\*y) ;**

**If (h<r1) and (h>r2) then**

**writeln('точка А лежит внутри кольца');**

**End.**

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел  $X$  и  $Y$ , при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой вариант).

**Решение**

1)  $x=0, y=2, r1=1, r2=3$  (В программе предполагается, что  $r1 > r2$ , хотя в условии это не оговаривается. Поэтому программа будет выдавать неверный ответ, когда  $r2 > r1$  и полученное расстояние  $h$  удовлетворяет условию  $r1 < h < r2$ ).

### **Program C1\_5;**

```
var x, y, r1, r2, h :real;
```

**Доработка программы:**

```
Begin
```

```
Write('Введите координаты точки');
```

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

```
Write('Введите радиусы двух окружностей');
```

```
readln (r1, r2);
```

```
h := sqrt(x*x + y*y);
```

```
if ((h < r1) AND (h > r2)) OR
```

```
  ((h < r2) AND (h > r1)) then
```

```
  writeln('точка A лежит внутри кольца');
```

```
Readln; end.
```

**C1 - 6.** Требовалось написать программу, которая решает неравенство

$$ax - b < 0$$

относительно **x** для любых чисел **a** и **b**, введенных с клавиатуры. Все числа считаются действительными.

Программист торопился и неправильно написал программу.

```
Program nerav2;  
Var a,b,x : real;  
Begin  
    readln(a,b,x);  
    if a=0 then  
        if b>0 then  
            write( 'любое число')  
            else  
                write( 'нет решений')  
        else  
            write( 'x < ',b/a);  
End.
```

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел  $a$ ,  $b$ ,  $x$ , при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.
- 3) 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).

**Решение**



- 1)  $a=-5$ ,  $b=10$ ,  $x=-3$  ( $a$  – любое отрицательное число)
- 2) Лишняя часть программы – использование и ввод переменной  $x$ .

## **Program C1\_6;**

Var a,b: real;

Begin

  readln(a,b);

  if a=0 then

    if b>0 then write( 'Любое число')

    else write( 'Нет решений')

  else

    if a>0 then write( 'x < ', b/a:8:2)

    else write( 'x > ', b/a:8:2);

  Readln;

End.

**Доработка программы:**

**C1-7.** Требовалось написать программу, которая определяет, лежит ли точка  $A(x_n, y_n)$  внутри треугольной области, ограниченной осями координат и прямой  $y = 2-x$  («внутри» понимается в строгом смысле, т.е. случай, когда т.А лежит на границе области, недопустим). В результате программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

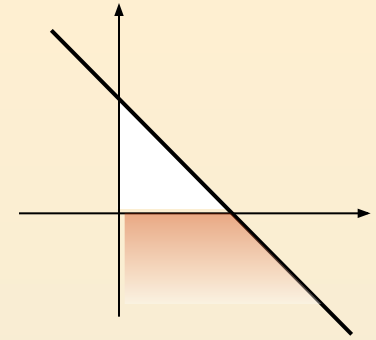
```
Var x0, y0, y : real;  
Begin  
readln(x0,y0);  
if (x0<2) then begin  
if (x0>0) then begin  
    y = 2 - x0;  
    if (y0 < y) then  
        writeln ( 'точка лежит внутри обл.')  
    else  
        writeln ( 'точка не лежит внутри обл.');  
    end  
    else  
        writeln ( 'точка не лежит внутри обл.');  
    End  
    else  
        writeln('точка не лежит внутри обл.');  
End.
```

## ***Последовательно выполните задания:***

- 1) Приведите пример таких чисел  $X_0$  и  $Y_0$ , при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев правильной работы (можно указать любой способ доработки исходной программы).

**Решение**

1)  $x=1, y=-1$  ( не учитывается условие  $y>0$  )



## Доработка программы:

### **Program C1\_7;**

```
var x0, y0, y: real;
```

```
Begin
```

```
Write('Введите координату точки');
```

```
readln (x0, y0);
```

```
if (x0 > 0) and (x0 < 2) and
```

```
    (y0 > 0) and (y0 < 2 - x0) then
```

```
    writeln ('точка лежит внутри области')
```

```
    else
```

```
    writein ('точка не лежит внутри области');
```

```
Readln;
```

```
End.
```

**C1-8.** Требовалось написать программу, которая решает неравенство  $(x - a)/(bx) > 0$  относительно  $x$  для любых ненулевых чисел  $a$  и  $b(a \neq 0, b \neq 0)$ , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу с ошибками:

```
Program trud1;  
Var a, b, x : Real;  
Begin  
    Readln(a, b, x);  
    If b>0 then  
        Write ('x>', a , 'или x<0')  
    else  
        If a>0 then  
            Write('0<x<', a)  
        else  
            Write( a, '<x<0');  
Readln  
End.
```

**Последовательно выполните задания:**

- 1) Приведите пример таких чисел **a, b, x**, при которых программа работает неправильно.
- 2) Укажите, какая часть программы является лишней.
- 3) Доработайте программу так, чтобы не было случаев ее неправильной работы . Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.

**Решение**



1) **a = -7, b = 12, x = 0** (*b – любое положительное число, a – любое отрицательное, x можно не указывать*).

2) Лишняя часть программы – ввод переменной **x**.

**Program C1\_8;**

Var a, b : real;

Begin

  Readln(a, b);

  If b>0 then

    If a>0 then Write ('x>', a:8:2, 'или x<0')

      else Write ('x<', a:8:2, 'или x>0')

      else

        If a>0 then Write('0<x<', a:8:2)

          else Write( a:8:2, '<x<0');

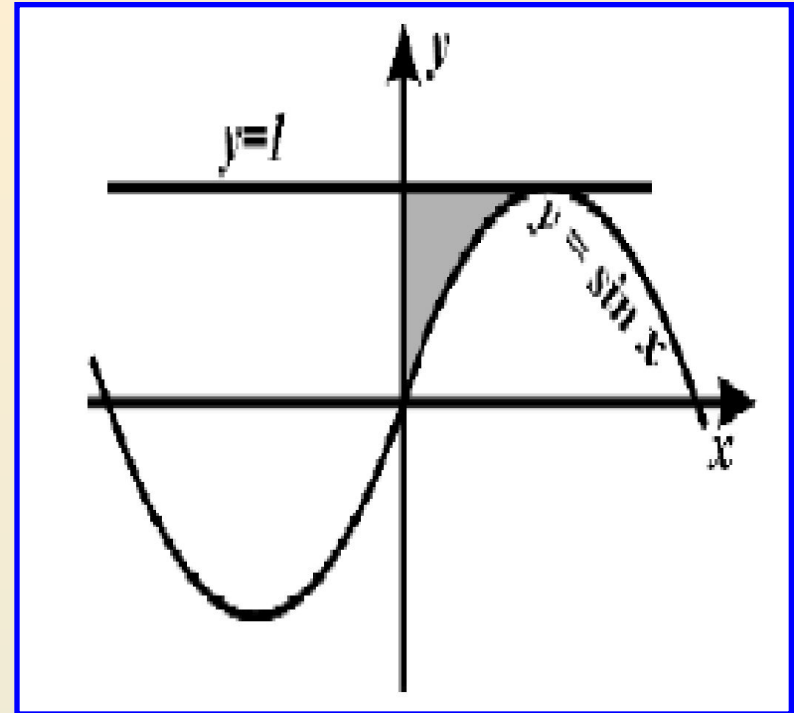
  Readln;

End.

**Доработка программы:**

**C1-9.** Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости  $(x, y)$  (действительные

и ~~виртуальные~~) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно.



```
Program tochka1;  
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');  
  readln  
end.
```

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел  $x$ ,  $y$  при которых программа работает неправильно.
- 2) Доработайте программу так, чтобы не было случаев ее неправильной работы. Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.

**Решение**

1. Программа работает неверно для точек:

**( $x=3.14$ ,  $y=0.5$ )** (*не верно определяет принадлежность точки области*)

**( $x=0$ ,  $y=1$ )** (*не выдает вообще никакого сообщения*).

## Доработка программы:

```
Program C1_9;
```

```
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.
```

**C1\_10.** Требовалось написать программу, которая решает неравенство  $(x + a)/(bx) < 0$  относительно  $x$  для любого ненулевого числа  $b$  ( $b \neq 0$ ) и любого неотрицательного числа  $a$  ( $a \geq 0$ ), введенных с клавиатуры. Все числа считаются действительными.

Программист торопился и написал программу с ошибками:

```
Program trud2;  
Var a, b, x : Real;  
Begin  
  Readln(a, b, x);  
  If a=0 then  
    If b>0 then Write('Решений нет')  
      else  
        Write('x>0 или x<0' )  
      else  
        Write( -a, '<x<0');  
Readln  
End.
```

**Последовательно выполните задания:**

- 1) Приведите пример таких чисел **a, b, x**, при которых программа работает неправильно.
- 2) Укажите, какая часть программы является лишней.
- 3) Доработайте программу так, чтобы не было случаев ее неправильной работы. Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.

**Решение**



- 1) **a=10, b=-2, x=0** (*a – любое положительное число, b – любое отрицательное, x можно не указывать*).
- 2) Лишняя часть программы – ввод переменной **x**.

### **Program C1\_10;**

Var a, b : real;

Begin

  Readln(a, b);

  If a=0 then

    If b>0 then Write('Решений нет')  
      else Write('x>0 или x<0' )

  else

    If b>0 then Write( -a:8:2, '<x<0')  
      else  
        Write('x>0 или x<',-a:8:2 );

  Readln

End.

**Доработка программы:**

**C1- 11.** Требовалось написать программу, которая решает уравнение  $ax^2+bx+c=0$  относительно  $x$  для действительных чисел  $a, b, c$ , введенных с клавиатуры, о которых заведомо известно, что  $a \neq 0, b \neq 0, c \neq 0$ .

Была написана следующая программа. Известно, что программа написана с ошибками.

```
Program korni;  
Var a, b, c, D, x1, x2 : real;  
Begin  
    readln(a,b,c,x1,x2);  
D:=b*b - 4*a*c;  
If (D>0) then  
    begin  
        x1:=(-b+sgrt(D))/(2*a);  
        x2:= (-b-sgrt(D) )/ (2*a) ;  
        write( 'x1 = ',x1, 'x2 = ',x2);  
    end  
    else  
writeln('действительных корней нет');  
End.
```

**Последовательно выполните задания:**

- 1) Приведите пример таких чисел ***a***, ***b***, ***c***, при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.
- 3) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы.

**Решение**

1) **a=1 b=2 c=1** (дискриминант равен нулю и должен быть корень, а программа выдает, что корней нет).

2) Лишняя часть программы – ввод переменных **x1, x2**.

### **Program C1\_11;**

```
var a, b, c, D, x1, x2: real;
```

```
begin
```

```
  readln(a, b, c);
```

```
  D := b*b - 4*a*c;
```

```
  if D >= 0 then begin
```

```
    x1 := (-b + sqrt(D)) / (2*a);
```

```
    x2 := (-b - sqrt(D)) / (2*a);
```

```
    writeln('x1 =', x1:8:2, 'x2 =', x2:8:2);
```

```
      end
```

```
  else writeln ('действительных корней нет');
```

```
  Readln;
```

```
end.
```

**Доработка программы:**

**C1- 12.** Требовалось написать программу, которая определяет, имеется ли среди введенных с клавиатуры целых чисел **a** и **b** хотя бы одно четное. Была написана следующая программа, в которой программист сделал ошибки:

**Program chet;**

**Var**

**a, b : integer;**

**Begin**

**readln(a,b);**

**a:=a mod 2;**

**If a>0 then b:=b mod 2;**

**If b>0 then**

**writeln('четных чисел нет')**

**else**

**writeln('четное число есть');**

**End.**

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел **a** и **b**, при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

**Решение**



1) **a=2, b=1** (первое число четное, а второе больше 0)

## Доработка программы:

```
Program C1_12;  
var a, b: integer;  
Begin  
write('Введите два числа');  
readln(a, b);  
if (a mod 2 = 0) or (b mod 2 = 0)  
then  
    writeln ('четное число есть')  
else  
    writeln ('четных чисел нет');  
Readln;  
end.
```

**C1-13.** Требовалось написать программу, которая решает следующее неравенство  $a > bx$  для любых чисел  $a$  и  $b$ .

Входная информация вводится с клавиатуры.

Программист торопился и написал программу некорректно:

```
Program ner;  
VAR a,b,x: real;  
BEGIN  
  readln(a,b,x);  
  if b=0 then  
    if a<=0 then  
      writeln('решения нет')  
    else  
      writeln('x любое число')  
    else  
      writeln(x,'<',a/b);  
readln  
END.
```

***Последовательно выполните задания:***

1. Приведите пример таких чисел  $a$ ,  $b$ ,  $x$ , при которых программа работает неправильно.
2. Укажите лишнюю часть программы и ошибки.
3. Доработайте программу так, чтобы не было случаев ее неправильной работы . Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.

**Решение**

- 1)  $a=5$ ,  $b=-5$ ,  $x=-3$  ( $b$  может быть любым отрицательным числом)
- 2) Лишняя часть программы – использование и ввод переменной  $x$ .

### **Program C1\_13;**

```
VAR a, b: real;  
BEGIN
```

**Доработка программы:**

```
  readln(a,b);  
  if b=0 then  
    if a<=0 then writeln('Решения нет')  
      else writeln('x - любое число')  
    else  
      if b>0 then writeln('x<',a/b:8:2)  
        else writeln('x>',a/b:8:2);  
Readln;  
END.
```

**C1-14.** Требовалось написать программу, которая решает следующее неравенство  **$ax < |b|$**  для любых чисел  $a$  и  $b$ .

Входная информация вводится с клавиатуры.

Программист торопился и написал программу некорректно:

## **Program neraven;**

```
VAR  a, b, x: real;
```

```
BEGIN
```

```
  readln(a,b,x);
```

```
  If a=0 then
```

```
    writeln('x любое число')
```

```
  else
```

```
    If a>0 then
```

```
      writeln(x,'<',b/a)
```

```
    else
```

```
      writeln(x,'>',b/a);
```

```
  readln
```

```
END.
```

***Последовательно выполните задания:***

1. Приведите пример таких чисел  $a$ ,  $b$ ,  $x$ , при которых программа работает неправильно.
2. Укажите лишнюю часть программы и ошибки.
3. Доработайте программу так, чтобы не было случаев ее неправильной работы . Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.

**Решение**



- 1)  $a=0$ ,  $b=0$ ,  $x=5$  ( $x$  – любое число, можно не указывать)
- 2) Лишняя часть программы – использование и ввод переменной  $x$ .

### **Program C1\_14;**

```
VAR  a,b: real;
```

```
BEGIN
```

```
  readln(a,b);
```

```
  If a=0 then
```

```
    if b=0 then writeln('решений нет')
```

```
      else writeln('x любое число')
```

```
      else
```

```
        If a>0 then writeln('x<',abs(b)/a:8:2)
```

```
          else writeln('x>',abs(b)/a:8:2);
```

```
  readln;
```

```
END.
```

**Доработка программы:**

**C1- 15.** Требовалось написать программу, которая определяет, лежит ли точка **A(x0,y0)** внутри квадратной области, ограниченной прямыми, параллельными осям координат : **y=-2, y=2, x=-2, x=2** («внутри» понимается в строгом смысле). В результате программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

```
Program tochka;  
Var x0,y0,y : real;  
Begin  
    readln(x0,y0);  
    If (x0<2) then begin  
        If (x0>-2) then  
            If (y0<2) then  
writeln('точка лежит внутри области');  
                end  
            else  
writeln('точка не лежит внутри области');  
End.
```

***Последовательно выполните задания:***

- 1) Приведите пример таких чисел  **$x_0$**  и  **$y_0$** , при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

**Решение**

1)  $x=0$ ,  $y=-7$  (точка в область не попала, а программа не выдает никаких сообщений).

## Доработка программы:

### **Program C1\_15;**

```
var x0, y0: real;
```

```
Begin
```

```
write('Введите координаты точки');
```

```
readln (x0, y0);
```

```
  if (x0 < 2) and (x0 > -2) and  
      (y0 < 2) and (y0 > -2)  then
```

```
      writeln('точка лежит внутри области')
```

```
      else
```

```
writeln('точка не лежит внутри области');
```

```
readln;
```

```
end.
```

**C1- 16.** Требовалось написать программу, которая определяет, можно ли построить треугольник из отрезков с длинами **x**, **y** и **z**. Программа должна выводить соответствующее текстовое сообщение.

Программист сделал в программе ошибки:

```
Program treug;  
Var x,y,z :real;  
Begin  
    Readln (x,y,z);  
    If (x+y>z) then begin  
        If (x+z>y) then  
            If (y+z>x) then  
writeln( 'треугольник построить можно');  
                end  
            else  
writeln('треугольник построить нельзя');  
End.
```

## ***Последовательно выполните задания:***

1. Приведите пример таких чисел  $x$ ,  $y$  и  $z$ , при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

**Решение**



**1)  $x=2, y=4, z=1$**  (не выдается никакого сообщения, если первое условие истинно ( $x + y > z$ ), а одно из других – ложно)

## Доработка программы:

### **Program C1\_16;**

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

```
Begin
```

```
Write('Введите длины трех отрезков');
```

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

```
if (x + y > z) and (x + z > y) and (y + z > x)
```

```
  then
```

```
    writeln('треугольник построить можно')
```

```
  else
```

```
    writeln('треугольник построить нельзя');
```

```
Readln;
```

```
end.
```

**C1- 17.** Окружность на плоскости задана координатами центра и радиусом. Даны две окружности. Требовалось определить, имеют ли они точку касания.

Программист поторопился и сделал в программе ошибки:

**Program okrug;**

**Var x1,y1,r1,x2,y2,r2 :real;**

**Begin**

**Readln (x1,y1,r1);**

**Readln (x2,y2,r2);**

**If sqrt(sqr(x2-x1)+sqr(y2-y1))=r1+r2**

**then**

**writeln( 'да')**

**else**

**writeln( 'нет');**

**End.**

***Последовательно выполните задания:***

1. Приведите пример таких исходных данных, при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

**Решение**

1)  $x_1=4, y_1=4, r_1=4, x_2=4, y_2=2, r_2=2$  (случай внутреннего касания)

## Доработка программы:

### **Program C1\_17;**

```
Var x1,y1,r1,x2,y2,r2 :real;
```

```
Begin
```

```
  Readln (x1,y1,r1);
```

```
  Readln (x2,y2,r2);
```

```
  If (sqrt(sqr(x2-x1)+sqr(y2-y1))=r1+r2) OR  
     (sqrt(sqr(x2-x1)+sqr(y2-y1))=abs(r1-r2))
```

```
  then writeln( 'да')
```

```
  else writeln( 'нет');
```

```
  Readln;
```

```
End.
```

**C1- 18.** Требовалось написать программу нахождения количества цифр в записи натурального числа  $A$ .

Программист поторопился и сделал в программе ошибки:

```
Program cifri;  
Var a,s :longint;  
Begin  
    write('Введите число');  
    Readln (A); s:=0;  
    while a >= 10 do  
        a:= a div 10;  
        s:=s+1;  
    writeln('Количество цифр:',s);  
readln  
End.
```

***Последовательно выполните задания:***

1. Приведите пример таких исходных данных, при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

**Решение**

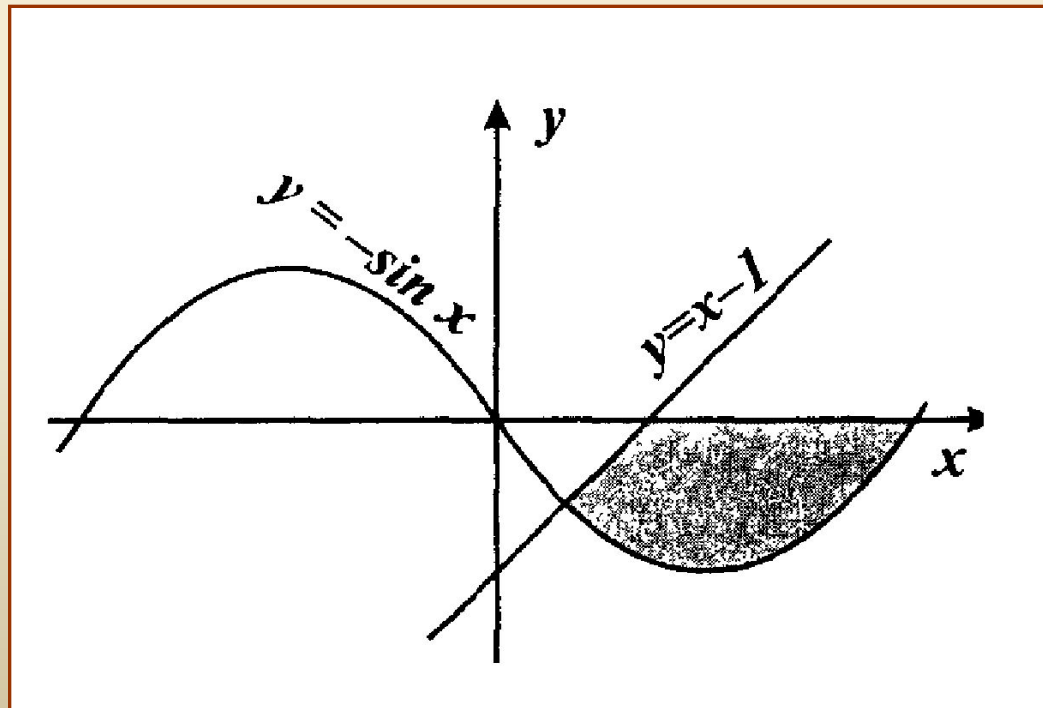


1) **a=124** (любое число большее 9)

## Доработка программы:

```
Program C1_18;  
Var a,s :longint;  
Begin  
    write('Введите целое число');  
    Readln(A); s:=0;  
    while a<>0 do begin  
        a:= a div 10;  
        s:=s+1;    end;  
    writeln('Количество цифр:',s);  
readln  
End.
```

**C1-19.** Составить программу, которая вводит с клавиатуры координаты точки на плоскости  $(x, y)$  и определяет принадлежность точки заштрихованной области, включая ее границы.



**VAR x, y : real;**

**Begin**

**readln ( x, y );**

**If Y <= x-1 then**

**if y <= 0 then**

**if y > - sin(x)**

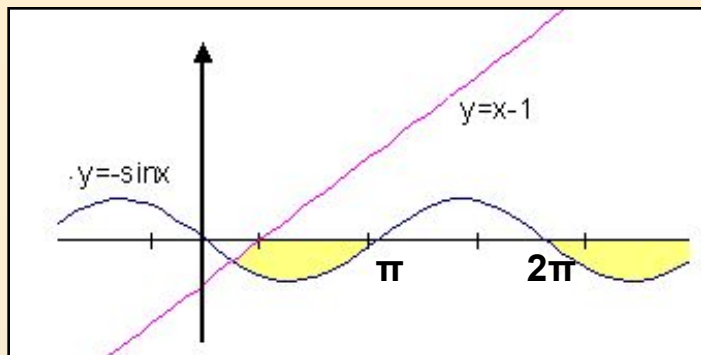
**then Writeln ('принадлежит')**

**else Writeln('не принадлежит');**

**end.**

**Найти  
ошибки в  
программе.**

**Решение**



Проверяемые в программе условия выделяют следующие области (закрашены желтым цветом).

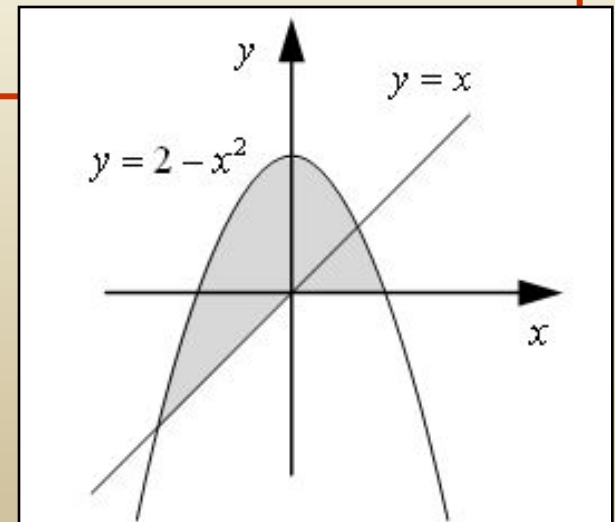
## Доработка программы:

1)  $x=3$  ,  $y=5$  {любая пара, для которой выполняются условия:  
 $(y > x-1)$  или  $(y > 0)$  или  
 $((y > -\sin x) \text{ и } (y \leq 0) \text{ и } (x \geq 2\pi.))$  }

### Program C1\_19;

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

**C1-20.** 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



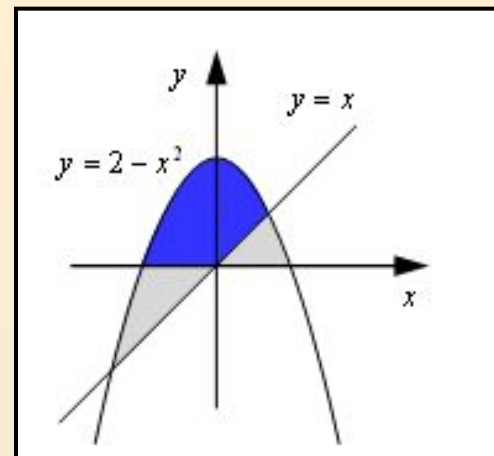
```
var x, y: real;  
begin  
  readln(x, y);  
  if y >= x then  
    if y >= 0 then  
      if y <= 2-x*x then  
        write('принадлежит')  
      else  
        write('не принадлежит');  
    end.
```

***Последовательно выполните задания:***

1. Приведите пример таких исходных данных, при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

**Решение**

1)  $x=0.5$ ,  $y=0.2$  (проверяемые условия выделяют только область, обозначенную на рисунке синим цветом, а не всю заштрихованную серую область. )



## Доработка программы:

### **Program C1\_20;**

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

```
begin
```

```
  write('Введите координаты точки');
```

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

```
  if ((y >= x) or (y >= 0)) and (y <= 2-x*x)
```

```
    then write('принадлежит')
```

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

```
  readln;
```

```
end.
```



# *Источники заданий*

- С.С. Крылов, П.А. Якушкин. ЕГЭ 2010. Информатика: сборник экзаменационных заданий. /ФИПИ — М.: Эксмо, 2009.
- П.А. Якушкин, Д.М. Ушаков. Самое полное издание типовых вариантов реальных заданий ЕГЭ 2010. Информатика. — М.: Астрель, 2009.
- О.В. Ярцева, Е.Н. Цикина. ЕГЭ-2009. Информатика. Самые новые задания. – М.: АСТ: Астрель, 2009.
- <http://kpolyakov.narod.ru>

***Евграфова Ольга Владимировна,***  
*учитель информатики МОУ гимназии №8 г.о. Коломна*  
*Московской области*