**Демонстрационный вариант ЕГЭ 2018 г. - задание №24** — На обработку поступает натуральное число, не превышающее 109. Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 5. Если в числе нет цифр, кратных 5, 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.

Бейсик

```
DIM N, DIGIT, MAXDIGIT AS LONG
INPUT N
MAXDIGIT = N MOD 10
WHILE N > 0
        DIGIT = N MOD 10
        IF DIGIT MOD 5 = 0 THEN
                IF DIGIT > MAXDIGIT THEN
                        MAXDIGIT = DIGIT
                END IF
        END IF
        N = N \ 10
WEND
IF MAXDIGIT = 0 THEN
        PRINT "NO"
ELSE
        PRINT MAXDIGIT
END IF
```

Алгоритмический язык

```
алг
нач
цел N, digit, maxDigit
ввод N
maxDigit := mod(N,10)
нц пока N > 0
        digit := mod(N ,10)
                если mod(digit, 5) = 0 то
                        если digit > maxDigit то
                                maxDigit := digit
                        все
                все
        N := div(N,10)
кц
если maxDigit = 0 то
        вывод "NO"
иначе
        вывод maxDigit
все
кон
```

C++

Python

```
N = int(input())
maxDigit = N % 10
while N > 0:
        digit = N % 10
        if digit % 5 == 0:
                if digit > maxDigit:
                        maxDigit = digit
        N = N // 10
if maxDigit == 0:
        print("NO")
else:
        print(maxDigit)
```

Паскаль

```
var N,digit,maxDigit: longint;
begin
readln(N);
maxDigit := N mod 10;
while N > 0 do
begin
        digit := N mod 10;
        if digit mod 5 = 0 then
                if digit > maxDigit then
                        maxDigit := digit;
        N := N div 10
end;
if maxDigit = 0 then
        writeln('NO')
else
        writeln(maxDigit)
end.
```

```
#include <iostream>
using namespace std;
int main() {
long N, digit, maxDigit;
cin >> N;
maxDigit = N % 10;
while (N > 0) {
        digit = N % 10;
        if (digit % 5 == 0)
                if (digit > maxDigit)
                        maxDigit = digit;
        N = N / 10;
}
if (maxDigit == 0)
        cout << "NO" << endl;
else
        cout << maxDigit << endl;
return 0;
}
```

Последовательно выполните следующее.
1. Напишите, что выведет эта программа при вводе числа 132.
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.

**Решение:**

Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков.
1. Программа выведет число 2.
2. Программа выдаёт правильный ответ, например, для числа 135.
Замечание для проверяющего. Программа работает неправильно из-за неверной начальной инициализации и неверной проверки отсутствия цифр, кратных 5. Соответственно, программа будет выдавать верный ответ,
если вводимое число содержит хотя бы одну цифру, кратную 5, и наибольшая цифра числа, кратная 5, не равна 0 и не меньше младшей (крайней правой) цифры числа (или просто стоит последней).
3. В программе есть две ошибки.
Первая ошибка. Неверная инициализация ответа (переменная maxDigit).
Строка с ошибкой:
maxDigit := N mod 10;
Верное исправление:
maxDigit := -1;
Вместо -1 может быть использовано любое число, меньшее 0.
Вторая ошибка. Неверная проверка отсутствия цифр, кратных 5.

Строка с ошибкой:
ifmaxDigit = 0 then
Верное исправление:
ifmaxDigit = -1 then
Вместо -1 может быть другое число, меньшее 0, которое было положено в maxDigit при исправлении первой ошибки, или проверка, что maxDigit< 0

---

**Демонстрационный вариант ЕГЭ 2017 г. - задание №24** — Дано целое положительное число N, не превосходящее 1000. Необходимо определить, является ли это число степенью числа 3. То есть требуется определить, существует ли такое целое число K, что $3^K = N$, 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. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

**Бейсик**

```
DIM N, K AS INTEGER
INPUT N
K = 0
WHILE K MOD 3 = 0
        K = K + 1
        N = N \ 3
WEND
IF N > 0 THEN
        PRINT K
ELSE
        PRINT "Не существует"
END IF
END
```

**Python**

```
n = int(input())
k = 0
while k%3 == 0:
        k = k + 1
        n = n // 3
if n > 0:
        print(k)
else:
        print("Не существует")
```

**Алгоритмический язык**

```
алг
нач
        цел n, k
        ввод n
        k := 0
```

```
        нц пока mod(k, 3)=0
                k := k + 1
                n := div(n,3)
        кц
        если n > 0
                то вывод k
                иначе вывод "Не существует"
        все
кон
```

**Паскаль**

```
var n, k: integer;
begin
        read(n);
        k := 0;
        while k mod 3 = 0 do begin
                k := k + 1;
                n := n div 3;
        end;
        if n > 0 then
                writeln(k)
        else
                writeln('Не существует')
end.
```

**Си**

```
#include <stdio.h>
int main(){
        int n, k;
        scanf("%d",&n);
        k = 0;
        while (k%3 == 0) {
                k = k + 1;
                n = n / 3;
        }
        if (n > 0)
                printf("%d", k);
        else
                printf("Не существует");
        return 0;
}
```

Последовательно выполните следующее.
1. Напишите, что выведет эта программа при вводе числа 9.
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.
Достаточно указать ошибки и способ их исправления для одного языка программирования.
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.

**Решение:**

1. При вводе числа 9 программа выведет число 1.
2. Примеры чисел, при вводе которых программа выводит корректный
ответ: 2, 3. Других чисел нет.
Комментарий для экспертов. После выполнения программы при любом введённом n значение k будет равно 1 (тело цикла выполнится ровно 1 раз).
В результате программа напечатает либо 1 (если n ≥ 3), либо «Не существует» (в противном случае). Таким образом, программа выводит корректный ответ, только если введено 2 или 3. Экзаменуемому достаточно
указать любое из этих чисел. Отметим, что при n=1 программа напечатает «Не существует», что неверно (должно быть напечатано «0»).
3. Программа содержит две ошибки:
1) неверное условие цикла;
2) неверное условие при печати результата.
**Пример исправления для языка Паскаль:**
**Первая ошибка:**
while k mod 3 = 0 do begin
Исправленная строка:
while n mod 3 = 0 do begin
**Вторая ошибка:**
if n>0 then
Исправленная строка:
if n=1 then
*Пояснение для эксперта*
После исправления первой ошибки в результате выполнения цикла значение переменной n будет равно $n0/(3^k)$, где n0 – введённое пользователем значение; k – максимальный показатель степени, при котором $3^k$ является делителем числа n0. Число n0 является степенью числа 3, если $n0 = 3^k$, т.е. $n0/(3^k) = 1$.

В программах на других языках ошибочные строки и их исправления аналогичны.
Незначительной опиской, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления

| Указания по оцениванию | Баллы |
|---|---|

| | |
|---|---|
| В задаче требуется выполнить три действия.<br>1. Указать результат программы при данном вводе.<br>Это действие считается выполненным, если указан верный результат работы программы при заданных входных данных. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.2. Указать пример ввода, при котором программа выводит верный<br>ответ.<br>Это действие считается выполненным, если указан пример числа, при вводе которого выводится верное сообщение (верный показатель степени или текст «Не существует», если введённое число не является степенью). Ученик не обязан указывать, что будет выведено, и объяснять, как работает программа.3. Найти и исправить ошибки в программе.<br>Это действие считается выполненным, если верно указаны обе ошибки и предложены верные варианты исправления, при этом никакие верные строки программы не указаны в качестве неверных. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).<br>Ошибка считается исправленной, если выполнены оба следующих условия:<br>а) правильно указана строка с ошибкой;<br>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа | |
| Выполнены все три необходимых действия, и ни одна верная строка не указана в качестве ошибочной | 3 |
| Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций.<br>1. Выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.<br>2. Выполнены два первых действия, найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.<br>3. Выполнено одно из первых двух действий, найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной | 2 |
| Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев.<br>1. Выполнены два первых действия. При этом несущественно, насколько правильно выполнено третье действие.<br>2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2.<br>3. Выполнено одно из двух первых действий. Исправлена одна из двух ошибок. Не более чем одна верная строка названа ошибочной | 1 |
| Не выполнены условия, позволяющие поставить 1, 2 или 3 балла | 0 |
| *Максимальный балл* | *3* |

# Демонстрационный вариант ЕГЭ 2016 г. – задание №24

На обработку поступает положительное целое число, не превышающее $10^9$. Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

### Бейсик

```
DIM N, DIGIT, SUM AS LONG
 INPUT N
 SUM = 0
 WHILE N > 0
    DIGIT = N MOD 10
    IF DIGIT < 7 THEN
        SUM = SUM + 1
    END IF
    N = N \ 10
 WEND
 PRINT DIGIT
```

### Python

```
N = int(input())
sum = 0
while N > 0:
    digit = N % 10
    if digit < 7:
        sum = sum + 1
    N = N // 10
print(digit)
```

### Алгоритмический язык

```
алг
 нач
    цел N, digit, sum
    ввод N
    sum := 0
    нц пока N > 0
        digit := mod(N,10)
        если digit < 7 то
            sum := sum + 1
        все
        N := div(N,10)
    кц
    вывод digit
 кон
```

### Паскаль

```
var N, digit, sum: longint;
 begin
    readln(N);
    sum := 0;
```

```
    while N > 0 do
    begin
        digit := N mod 10;
        if digit < 7 then
            sum := sum + 1;
        N := N div 10;
    end;
    writeln(digit)
 end.
```

**Си**

```
#include <stdio.h>
 int main()
 {
    int N, digit, sum;
    scanf("%d", &N);
    sum = 0;
    while (N > 0)
    {
        digit = N % 10;
        if (digit < 7)
            sum = sum + 1;
        N = N / 10;
    }
    printf("%d",digit);
    return0;
 }
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 456.
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.

**Решение:**

Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков.
1. Программа выведет число 4.
2. Пример числа, при вводе которого программа выдаёт верный ответ: 835.
*Замечание для проверяющего. Программа работает неправильно из-за неверной выводимой на*

*экран переменной и неверного увеличения суммы.*
*Соответственно, программа будет работать верно, если в числе старшая цифра (крайняя левая) равна сумме цифр, меньших 7.*
3. В программе есть две ошибки.
Первая ошибка. Неверное увеличение суммы.
Строка с ошибкой:
*sum := sum + 1;*
Верное исправление:
*sum := sum + digit;*
Вторая ошибка. Неверный вывод ответа на экран.
Строка с ошибкой:
*writeln(digit)*
Верное исправление:
*writeln(sum)*

---

## ЕГЭ 16.06.2016 по информатике. Основная волна. Вариант 41 (Часть С)

Ученик написал программу, определяющую, какой степенью числа 4 является введенное. Например, для 16 это 2, так как $4^2 = 16$. Если же такой степени нет, то необходимо вывести сообщение «Не существует». К сожалению, ученик написал программу неверно.

**Паскаль**

```
var n,k: longint;
begin
readln(n);
k := 0;
while k mod 4 = 0 do
begin
        n := n div 4;
        k := k + 1;
end;
if n <= 4 then
        writeln(k)
else
writeln('Не существует');
end.
```

Последовательно выполните следующее.

1. Что выдаст программа при вводе числа 64?

2. При вводе какого числа программа выдаст верный ответ? Укажите 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. Помните, что нужно исправить имеющуюся программу, а не писать свою, хоть

24. Поиск и исправление ошибок в программе

и с лучшим алгоритмом.

**Решение:**

Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков.
1. Программа выведет число 3.
2. Пример числа, при вводе которого программа выдаёт верный ответ: 64.
3. В программе есть две ошибки.
Первая ошибка.
Строка с ошибкой:
while k mod 4 = 0 do
Верное исправление:
while n mod 4 = 0 do
Вторая ошибка.
Строка с ошибкой:
if n <= 4 then
Верное исправление:
if n = 1 then

---

Требовалось написать программу, которая определяет, лежит ли точка $A(x_0,y_0)$ внутри треугольной области, ограниченной осями координат и прямой **y=2-x** («внутри» понимается в строгом смысле, т.е. случай, когда точка А лежит на границе области, недопустим). В результате программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

**Паскаль**

```
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 области')
                else writein ('точка не лежит внутри области');
        end
        else writeln ('точка не лежит внутри области');
end
else writeln ('точка не лежит внутри области');
end.
```

**Си**

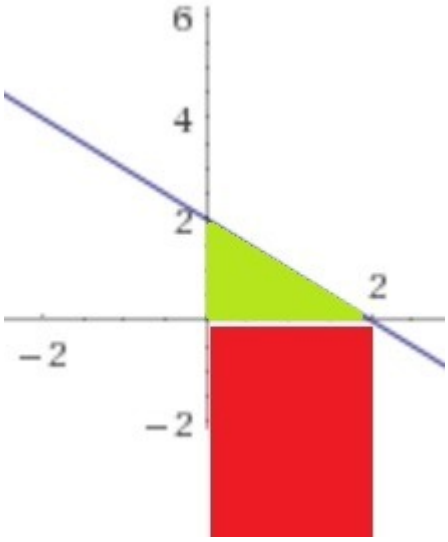
```
#include <stdio.h>
int main(void)
{ float x0, y0, y;
scanf (″%f %f″,x0, y0);
if (x0 < 2) {
```

```
        if (x0 > 0) {
                y = 2 – x0;
                if (y0 < y)
                        printf (″точка лежит внутри области″);
                else printf (″точка не лежит внутри области″);
        }
        else printf (″точка не лежит внутри области″);
}
else printf (″точка не лежит внутри области″);
}
```

Последовательно выполните задания:

Приведите пример таких чисел $x_0$ и $y_0$, при которых программа неверно решает поставленную задачу.
Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).
Укажите, как можно доработать программу, чтобы вместо вложенных операторов **IF** она содержала логическую операцию **AND**.

**Решение:**




1. (-1,1);(-2,1);(-3,1)

2.

**Паскаль**

```
...
if (x0 > 0)then begin
  if (y0 > 0)then begin
    y := 2 – x0;
    if (y0 < y) then
         writeln ('точка лежит внутри области')
    else writein ('точка не лежит внутри области');
  end
  else writeln ('точка не лежит внутри области');
end
```

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

**Си**

```
...
if (x0 > 0)  {
        if (y0 > 0)  {
                y = 2 – x0;
                if (y0 < y)
                        printf (″точка лежит внутри области″);
                else printf (″точка не лежит внутри области″);
                }
        else printf (″точка не лежит внутри области″);
        }
else printf (″точка не лежит внутри области″);
}
```

3.

```
if (x0 > 0) and (y0 > 0) and (y0 < 2 - x0) then begin
     writeln ('точка лежит внутри области')
else writein ('точка не лежит внутри области');
```

```
if (x0 > 0 && y0 > 0 && y0 < 2 - x0) )
        printf (″точка лежит внутри области″);
else
        printf (″точка не лежит внутри области″);
```

---

Требовалось написать программу, которая определяет, можно ли построить треугольник из отрезков с длинами **x**, **y**, **z**. Программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

```
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
```

```
#include <stdio.h>
 int main(void)
{ float x, y, z;
scanf (″%f %f %f″, &x, &y, &z);
if (x + y > z)
{
```

```
if (x + z > y)
if (y + z > x)
printf(″треугольник построить можно″);
}
else printf(″треугольник построить нельзя″);
}
```

Последовательно выполните задания:

Приведите пример таких чисел **x**, **y**, **z**, при которых программа неверно решает поставленную задачу.
Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать один из способов доработки исходной программы).
Укажите, как можно доработать программу, чтобы она вместо вложенных операторов **IF** содержала логическую операцию **AND**.

**Решение:**

1. Треугольник можно построить, если длина каждой стороны меньше суммы длин двух оставшихся, это условие нужно проверить для все трех сторон.

x=5, y=3, z=1

if (y + z > x) : if 3 + 1 > 5): Для этого ‘if’ нет ‘else’, поэтому он ничего не пишет.

2.

```
if (x + y > z) then begin
  if (x + z > y) then
    if (y + z > x) then
         writeln('треугольник построить можно')
    else writeln('треугольник построить нельзя')
  else writeln('треугольник построить нельзя');
end
else writeln('треугольник построить нельзя');
end.
```

```
if (x + y > z)
{
        if (x + z > y)
                if (y + z > x)
                        printf(″треугольник построить можно″);
                else printf(″треугольник построить нельзя″);
        else printf(″треугольник построить нельзя″);
}
else printf(″треугольник построить нельзя″);
}
```

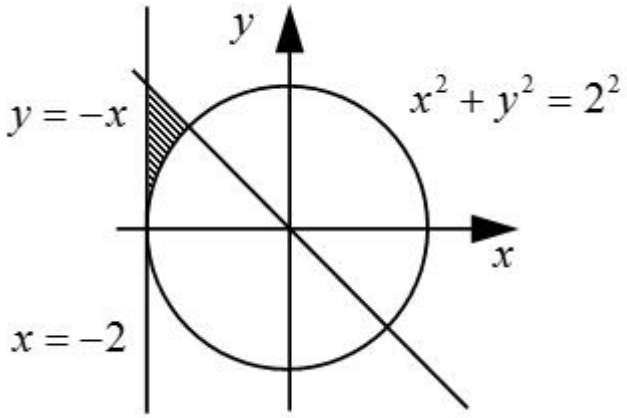
3.

```
if (x + y > z) and (x + z > y) and (y + z > x) then
     writeln('треугольник построить можно')
else writeln('треугольник построить нельзя');
```

```
if (x + y > z && x + z > y && y + z > x)
        printf(″треугольник построить можно″);
else printf(″треугольник построить нельзя″);
```

---

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 x*x + y*y >= 4 then
        if x >= –2 then
                if y <= –x then
                        write('принадлежит')
else
        write('не принадлежит')
end.
```

```
#include <stdio.h>
int main(void)
{ float x, y;
scanf(″%f %f″, &x, &y);
if (x*x + y*y >= 4)
        if (x >= –2)
                if (y <= –x)
                        printf(″принадлежит″);
else
        printf(″не принадлежит″);
}
```

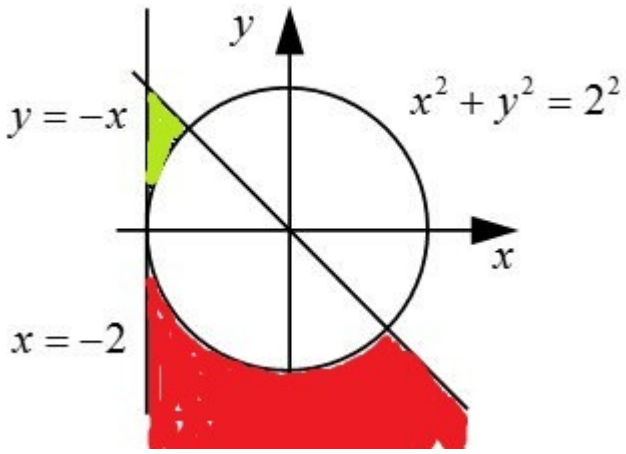
Последовательно выполните следующее:

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).

**Решение:**




1. (-1,-3); (1,3)

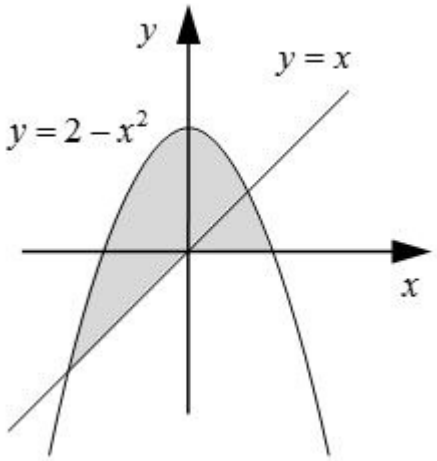
2.

```
if (x*x+y*y >= 4) and (x >= -2) and
   (y <= -x) and (y >= 0) then
  write('принадлежит')
else
  write('не принадлежит')
```

```
if (x*x + y*y >= 4 && x >= –2 && y <= –x &&  y >= 0)
        printf(″принадлежит″);
else
        printf(″не принадлежит″);
```

---

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



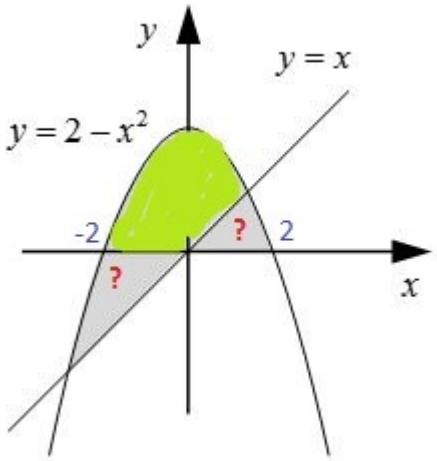

Программист торопился и написал программу неправильно. Вот она:

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

```
#include <stdio.h>
int main(void)
{
float x, y;
scanf("%f %f", &x, &y);
if (y >= x)
        if (y >= 0)
                if (y <= 2-x*x)
                        printf("принадлежит");
                else
                        printf("не принадлежит");
}
```

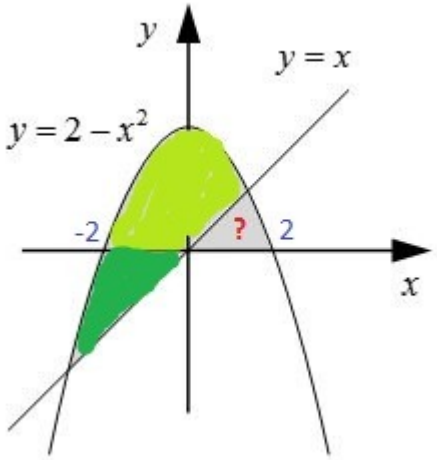
Последовательно выполните следующее:
1) Приведите пример таких чисел **x**, **y**, при которых программа неверно решает поставленную задачу.
2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой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

**Решение:**

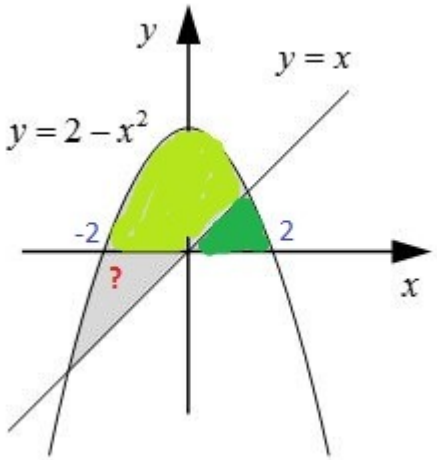



1. (1,0.5); (-1,-0.5)

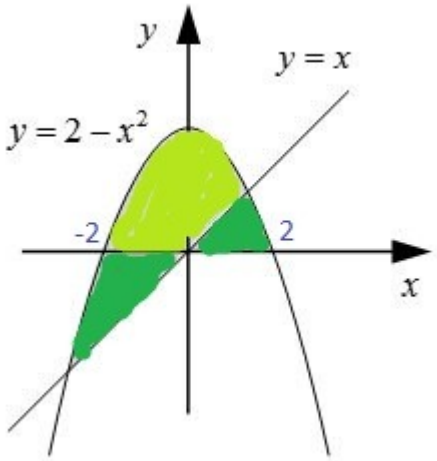
2. (y >= x) and (y <= 2-x*x)




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




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

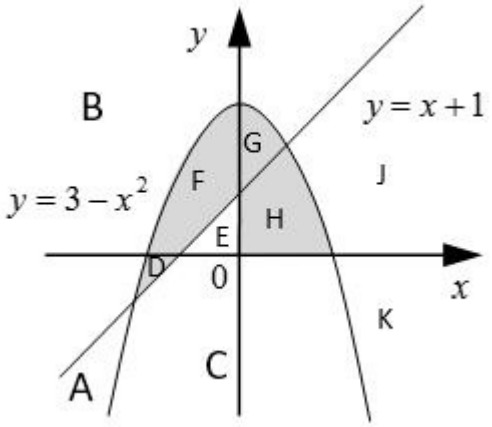



```
if ((y >= x) or (y >= 0)) and (y <= 2-x*x) then
    write('принадлежит')
  else
    write('не принадлежит');
```

```
if (((y >= x) || (y >= 0)) && (y <= 2-x*x))
        printf("принадлежит");
else
        printf("не принадлежит");
```

---

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 <= 3-x*x then
if y >= 0 then
if y >= x+1 then
```

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

```
#include <stdio.h>
int main(void)
{ float x, y;
scanf("%f %f", &x, &y);
if (y <= 3-x*x)
if (y >= 0)
if (y >= x+1)
printf("принадлежит");
else
printf("не принадлежит");
}
```

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G, H). Точки, лежащие на границах областей, отдельно не рассматривать.

| Область | y <= 3-x*x | y >= 0 | y >= x+1 | вывод | верно? |
|---|---|---|---|---|---|
| A | | | | | |
| B | | | | | |
| C | | | | | |
| D | | | | | |
| E | | | | | |
| F | | | | | |
| G | | | | | |
| H | | | | | |
| J | | | | | |
| K | | | | | |

В столбцах условий укажите «да», если условие выполнится, «нет» если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв». В последнем столбце укажите «да» или «нет».

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).

**Решение:**

1.

| Область | y <= 3-x*x | y >= 0 | y >= x+1 | вывод | верно? |
|---|---|---|---|---|---|
| A | нет | – | – | – | нет |
| B | нет | – | – | – | нет |
| C | да | нет | – | – | нет |
| D | да | нет | – | – | нет |
| E | да | да | нет | не принадлежит | да |
| F | да | да | да | принадлежит | да |
| G | да | да | да | принадлежит | да |
| H | да | да | нет | не принадлежит | нет |
| J | нет | – | – | – | нет |
| K | нет | – | – | – | нет |

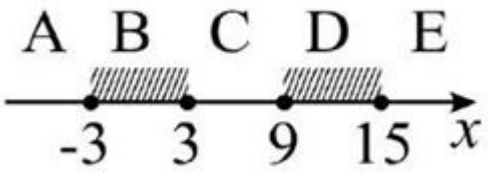
2.

```
if (y <= 3-x*x) and ((y >= x+1) or (x>=0) and (y>=0)) then
        write('принадлежит')
else
        write('не принадлежит');
```

```
if (y <= 3-x*x) && ((y >= x+1) || (x>=0 && y >= 0))
        printf(″принадлежит″);
else
        printf(″не принадлежит″);
```

---

Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (*x* – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы). Программист торопился и написал программу неправильно.




```
var x: real;
begin
  readln(x) ;
  if x<=l5 then
```

```
    if x<9 then
      if x<=3 then
        write('принадлежит')
      else
        write('не принадлежит')
end.
```

```
#include <stdio.h>
 int main(void)
{ float x;
  scanf(″%f″, &x);
  if (x<=l5)
    if (x<9)
      if (x<=3)
        printf(″принадлежит″);
      else
        printf(″не принадлежит″);
}
```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки -3, 3, 9 и 15) принадлежат заштрихованным областям.

| Область | (x<=15) | (x<9) | (x<=3) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| A | | | | | |
| B | | | | | |
| C | | | | | |
| D | | | | | |
| E | | | | | |

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите «Да» или «Нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

**Решение**

1.

| Область | (x<=15) | (x<9) | (x<=3) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| A | да | да | да | принадлежит | нет |
| B | да | да | да | принадлежит | да |
| C | да | да | нет | не принадлежит | да |
| D | да | нет | – | – | нет |
| E | нет | – | – | – | нет |

2.

```
if (-3 <= x) and (x <= 3) or (9 <= x) and (x <= 15) then
    write('принадлежит')
else
        write('не принадлежит');
```

```
if (-3 <= x && x <= 3) || (9 <= x && x <= 15)
        printf(″принадлежит″);
else
        printf(″не принадлежит″);
```