

Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
«Высшая школа экономики»

Московский институт электроники и математики Национального  
исследовательского университета «Высшая школа экономики»

Департамент прикладной математики

## ОТЧЕТ

По лабораторной работе №2  
«Работа 2. Определение принадлежности точки заданной области»  
По курсу «Алгоритмизация и программирование»

ФИО	Номер группы	Дата
Цыплаков Александр Александрович	БПМ214	04.10.2021

Москва-2021

## Вариант 25

### Задание:

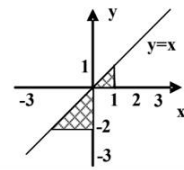
Даны числа  $x$  и  $y$ . Определить, принадлежит ли точка с координатами  $(x, y)$  заштрихованной области, включая границы.

Оформить первое решение в виде вложенных условных операторов с простыми условиями.

Второе решение должно содержать один условный оператор со сложным логическим условием.

Третье решение должно быть оформлено в виде отдельной функции, вызываемой из основной программы. Функция не содержит условного оператора, а только логическое выражение.

25



### 1)Решение:

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <locale.h>
```

```
int uslovie3(double x, double y) // создаем функцию, которая будет вызываться в 3 условия
```

```
{
    return (((x >= 0) && (x <= 1) && (x >= y)) || ((x >= -2) && (y <= 0) && (y >= -2) && (x >= y))); // используем для
    возврата значений
}
```

```
int main() // начало главной функции
```

```
{
    double x, y; // объявляем числа с плавающей запятой двойной точности
    setlocale(LC_ALL, "rus"); // подключаем русский язык
    setlocale(LC_NUMERIC, "eng"); // для того, чтобы мы могли вводить числа с десятичной точкой, а не запятой
    printf("Введите значения x и y\n"); // строка, которая будет выводиться в консольном окне
    scanf("%lf%lf", &x, &y); // форматированный ввод вещественных чисел
}
```

```
int a = 0; // инициализируем целочисленную переменную
```

```
//УСЛОВИЕ 1
```

```
if (x >= 0) // начало условного оператора с простыми условиями
```

```
{
    if (x <= 1) // второе условие
```

```
{
    if (x >= y) // третье условие
```

```
    a = 1; // получаемый результат этого пути условного оператора
```

```
}
}
```

```
else // другой путь условного оператора
```

```
{
    if (x >= -2) // начало условного оператора с простыми условиями
```

```
{
    if (y <= 0) // второе условие
```

```
{
    if (y >= -2) // третье условие
```

```
{
    if (x >= y) a = 1; // четвертое условие и результат, получаемый, если программа будет идти по этому пути
```

```
    else a = 0; // в случае, если предыдущие два пути не подходят результатом будет a = 0
```

```
}
}
```

```
}
```

```

    }
}

if (a == 1) // новый условный оператор, необходимый для выбора нужной строки
{
    printf("УСЛОВИЕ 1: принадлежит\n"); // строка, выводимая в консольное окно, если результатом будет a = 1
}
else printf("УСЛОВИЕ 1: не принадлежит\n"); // строка, выводимая в консольное окно, если результатом будет a = 0

//УСЛОВИЕ 2
if (((x >= 0) && (x <= 1) && (x >= y)) || ((x >= -2) && (y <= 0) && (y >= -2) && (x >= y))) // условный оператор со
сложным логическим условием
{
    printf("УСЛОВИЕ 2: принадлежит\n"); // строка, выводимая в консольное окно, если координата удовлетворяет условиям
}
else printf("УСЛОВИЕ 2: не принадлежит\n"); // строка, выводимая в консольное окно, если координата не удовлетворяет условиям

//УСЛОВИЕ 3
if (uslovie3(x, y) == 1) // вызываем функцию uslovie3, которую написали перед функцией main
{
    printf("УСЛОВИЕ 3: принадлежит\n"); // строка, выводимая в консольное окно, если координата удовлетворяет условиям
}
else printf("УСЛОВИЕ 3: не принадлежит\n"); // строка, выводимая в консольное окно, если координата не удовлетворяет условиям
return 0;
}

```

`#define _CRT_SECURE_NO_WARNINGS` – эта строка нужна, чтобы нам не отображалась ошибка о небезопасности кода, так как некоторые старые названия функций могут считаться небезопасными и либо можно их заменять на безопасные, либо прописать эту строчку в начале

`#include <stdio.h>` - библиотека, содержащая определения ввода и вывода

`#include <math.h>` - тут содержатся все математические функции, используемые в ходе программы

`#include <locale.h>` - библиотека, используемая для задач, связанных с локализацией. Одна из ключевых функций, которую представляет данный заголовочный файл – `setlocale`

После подключения всех библиотек начинается функция `uslovie3`, к которой мы будем возвращаться в третьем условии.

Далее начинается главная функция `int main`.

- `double x, y;` - объявляем числа с плавающей запятой двойной точности

- `setlocale(LC_ALL, "rus");` - подключаем русский язык
- `setlocale(LC_NUMERIC, "eng");` - для того, чтобы мы могли вводить числа с десятичной точкой, а не запятой
- `printf("Введите значения x и y\n");` - строка, которая будет выводиться в консольном
- `scanf("%lf%lf", &x, &y);` - форматированный ввод вещественных чисел
- `int a = 0;` - инициализируем целочисленную переменную со значением 0
- `if { if{ if}}` – условный оператор с простыми условиями
- `a = 1` – в случае если выполняются все условия присваиваем значение
- `else` – другой путь по которому пойдет программа в случае не выполнения хотя бы одного условия
- `if { if{ if{ if }}` условные операторы с простыми условиями
- `a = 1` - в случае если выполняются все условия присваиваем значение 0
- `else a = 0` – в случае если хотя бы одно условие не выполнится присваиваем значение 1
- `if ( a == 1)` – новый условный оператор необходимый для выбора нужной выводимой строки
- `printf("УСЛОВИЕ 1: принадлежит\n")` – выводимая строка в случае принадлежности точки к одной из областей по первому условию
- `else printf("УСЛОВИЕ 1: принадлежит\n")` – выводимая строка в случае, если точка не принадлежит ни одной из областей по первому условию
- `if (((x >= 0) && (x <= 1) && (x >= y)) || ((x >= -2) && (y <= 0) && (y >= -2) && (x >= y)))` – условный оператор содержащий логическое выражение
- `printf("УСЛОВИЕ 2: принадлежит\n")` - выводимая строка в случае принадлежности точки к одной из областей по второму условию
- `printf("УСЛОВИЕ 2: принадлежит\n")` – выводимая строка в случае, если точка не принадлежит ни одной из областей по второму условию
- `if (uslovie3(x, y) == 1)` – вызов функции, которую написали перед главной функцией, где прописано логическое выражение, в случае истинности присваивается значение 1
- `printf("УСЛОВИЕ 3: принадлежит\n");` - выводимая строка, если координата принадлежит хотя бы одной области по третьему условию
- `else printf("УСЛОВИЕ 3: принадлежит\n");` - выводимая строка, если координата не принадлежит ни одной области по третьему условию

Тело функции состоит из трех вариантов решений одной задачи: определение принадлежности точки заданной области

1)Первое решение в виде вложенных условных операторов с простыми условиями.

Используем условный оператор `if`. Прописываем условия задачи: в моем случае – две области под графиком ( $y=x$ ). Для первой области прописываем все условия, используя 3 условных оператора `if`: `((x >= 0); (x <= 1); (x >= y))`, после этого пишем `else` и прописываем все условия для второй области, используя 5 условных операторов `if`: `((x <= 0); (x >= -2); (y <= 0); (y >= -2); (x >= y))` – если после прогона введенных значений через условные операторы, они удовлетворяют всем условиям, то присваиваем объявленной переменной значение ( $a = 1$ ), если координата не удовлетворяет хотя бы одному условию, то присваиваем значение ( $a = 0$ ). Далее, в зависимости от присвоенного значения, выводим строку, используя `printf`, «УСЛОВИЕ 1: принадлежит/ не принадлежит»

2) Второе решение должно содержать один условный оператор со сложным логическим условием.

В этом случае прописываем все условия в одном условном операторе `if`, объединяя их логическими знаками (`&&` - и, `||` - или). Как и в первом варианте решения, если координаты удовлетворяют всем условиям, то выводим строку «УСЛОВИЕ 2: принадлежит», если не удовлетворяет хотя бы одному, выводим «УСЛОВИЕ 2: не принадлежит»

3) Третье решение должно быть оформлено в виде отдельной функции, вызываемой из основной программы. Функция не содержит условного оператора, а только логическое выражение.

Над главной функцией `main`, написана еще одна функция `uslovie3`, где содержится только логическое выражение. В главной функции `main`, мы вызываем функцию `uslovie3`, с помощью условного оператора `if`. Как и в первых двух случаях, если координаты удовлетворяют всем введенным значениям, выводим строку «УСЛОВИЕ 3: принадлежит», если не удовлетворяет хотя бы одному - «УСЛОВИЕ 3: не принадлежит»

## 2)Тесты:

### Тесты №1

```
Введите значения x и y
0.211 0.123
УСЛОВИЕ 1: принадлежит
УСЛОВИЕ 2: принадлежит
УСЛОВИЕ 3: принадлежит
```

### Тест №2

Введите значения  $x$  и  $y$   
2 3.5  
УСЛОВИЕ 1: не принадлежит  
УСЛОВИЕ 2: не принадлежит  
УСЛОВИЕ 3: не принадлежит

### Тест №3

Введите значения  $x$  и  $y$   
-0.21 -0.34  
УСЛОВИЕ 1: принадлежит  
УСЛОВИЕ 2: принадлежит  
УСЛОВИЕ 3: принадлежит