МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 002

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Лаптев

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

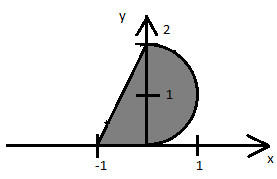
Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2019

**ЗАДАНИЕ №1.**

**1. Формулировка задачи**

Создать программу, которая позволяет определить, попадает ли точка с заданными координатами x и y в заштрихованную область.



**2. Постановка задачи**

На вход программы поступают два вещественных числа, являющиеся координатами (x, y) точки на плоскости. По этим координатам проверяются условия принадлежности/непринадлежности данной точки частям плоской фигуры. На выход поступает письменное сообщение о том, попадает или не попадает точка в заштрихованную область.

**3. Математическая модель**

На вход программы поступают два вещественных числа, являющиеся координатами точки (x, y). По этим координатам проверяется условие их принадлежности отрезкам, в которых находиться заштрихованная область (в данном случае от -1 до 1 – по оси х, и от 0 до 2 – по оси у), затем по координатам точки определяется, попадает ли она в заштрихованную область полукруга (уравнение окружности для которого: y = x2 + (y – 1)2), если она туда не попадает, то при помощи данной наклонной прямой, являющейся гипотенузой прямоугольного треугольника (уравнение для которой: y = 2 \* x + 2), построенного на осях координат и в четвертой четверти координатной плоскости, находится параллельная ей прямая, у которой проверяется условие нахождения её в пространстве относительно данной прямой. Если полученная прямая находится ниже заданной, то точка попадает в заштрихованную область, при условии, что прямая не находиться ниже графика функции и выполнении остальных ограничений на точку, связанных с её непопаданием в границы отрезков, на которых определён график данной функции, которые были описаны выше.

**4. Описание алгоритма**

Начало

4.1 Ввод координат x, y

4.2 Проверка условия, при котором координата будет попадать в заданное множество возможных значений для треугольника

4.3 Проверка условия, при котором координата будет попадать непосредственно внутрь заштрихованного треугольника или на его контур

4.4 Проверка условия, при котором точка не будет находиться в заданном множестве значений для треугольника

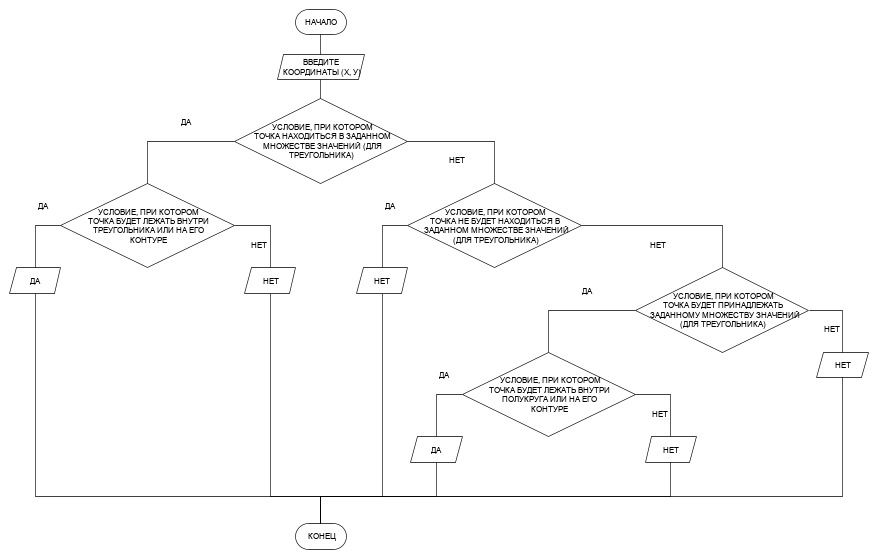
4.5 Проверка условия, при котором точка будет принадлежать заданному множеству значений для полукруга

4.6 Проверка условия, при котором точка будет попадать внутрь полукруга или на его контур

4.7 Вывод ответа

Конец

**5. Опорный граф (блок-схема) алгоритма**



**6. Проект программы с определением замкнутых программных единиц и необходимых структур данных**

В данном случае программа должна состоять из трех частей – блок ввода, блок вывода и блок преобразований, производимых над заданной точкой и проверки условий, связанных с вхождением/не вхождением точки в заштрихованную область. В качестве структуры данных могут быть введены переменные для хранения исходных координат (x, y).

**7. Текст программы:**

**на языке С++**

#include <iostream>;

#include <math.h>;

using namespace std;

int main()

{

float x, y;

setlocale(0, "");

cout << "Введите координаты точки: ";

cin >> x >> y; // Введение координат

if ((x >= -1) && (x <= 0) && (y >= 0) && (y <= 2)) // Условие, при котором точка будет находиться в заданном множестве значений (для треугольника)

{

if (y <= 2 \* x + 2) // Условие, при котором точка будет лежать внутри треугольника или на его контуре (в заданном множестве значений)

cout << "Да" << endl;

else

cout << "Нет" << endl;

}

else if ((x < -1) && (x > 0) && (y < 0) && (y > 2)) // Условие, при котором точка не будет находиться в заданном множестве значений (для треугольника)

cout << "Нет" << endl;

else if ((x >= 0) && (x <= 1) && (y >= 0) && (y <= 2)) // Условие, при котором точка будет принадлежать заданному множеству значений (для полукруга)

{

if (x \* x + (y - 1) \* (y - 1) <= 1) // Условие, при котором точка будет находиться внутри полукруга или на его контуре (в заданном множестве значений)

cout << "Да" << endl;

else

cout << "Нет" << endl;

}

else

cout << "Нет" << endl;

return 0;

}

**на языке Pascal**

**Program** lab\_2\_1;

**Var**

x, y:real;

**Begin**

Writeln('Введите координаты точки: ');

Readln(x, y); // Введение координат

**if** ((x >= -1) **and** (x <= 0) **and** (y >= 0) **and** (y <= 2)) **then** // Условие, при котором точка будет находиться в заданном множестве значений (для треугольника)

**begin**

**if** (y <= 2 \* x + 2) **then** // Условие, при котором точка будет лежать внутри треугольника или на его контуре (в заданном множестве значений)

Writeln('Да')

**else**

Writeln('Нет');

**end**

**else if** ((x < -1) **and** (x > 0) **and** (y < 0) **and** (y > 2)) **then** // Условие, при котором точка не будет находиться в заданном множестве значений (для треугольника)

Writeln('Нет')

**else if** ((x >= 0) **and** (x <= 1) **and** (y >= 0) **and** (y <= 2)) **then** // Условие, при котором точка будет принадлежать заданному множеству значений (для полукруга)

**begin**

**if** (x \* x + (y - 1) \* (y - 1) <= 1) **then** // Условие, при котором точка будет находиться внутри полукруга или на его контуре (в заданном множестве значений)

Writeln('Да')

**else**

Writeln('Нет');

**end**

**else**

Writeln('Нет');

**End**.

**8. Проверка работоспособности (тестирование) программы**

Для проверки работоспособности по запросу (после запуска программы) введем числа x = 1, y = 1. Программа должна выдавать в результате ответ: да. Введем числа x = 1, y = 2. Программа должна выдавать в результате ответ: нет. Введем числа x = -0,5, y = 1. Программа должна выдавать в результате ответ: да. Введем числа x = -1, y = 1. Программа должна выдавать в результате ответ: нет.

**9. Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду своей простоты программа одинаково быстро выполняется на Pascal и C++.

**ЗАДАНИЕ №2.**

**1. Формулировка задачи**

С клавиатуры вводятся три попарно различных действительных числа. Если их сумма меньше 1, то наименьшее из чисел заменяется полусуммой двух других. Иначе меньшее из первых двух чисел заменить на полусумму оставшихся.

**2. Постановка задачи**

На вход программы поступают три попарно различных действительных числа. Вычисляется их общая сумма. Проверяется место на числовой прямой этой суммы относительно 1. По итогам этой проверки выполняется ряд условий. На выходе программы выводятся три полученных в результате предыдущих действий вещественных числа.

**3. Математическая модель**

На вход программы поступает три попарно различных действительных числа (x, y, z). Находится сумма этих чисел. Сначала проверяется условие, при котором три числа не будут равны между собой. И если они вдруг равны, то программа прекращает свою работу. При помощи неравенства вычисляется положение этой суммы относительно 1 на числовой прямой. Если сумма меньше 1, тогда меньшее из данных чисел заменяется полусуммой оставшихся двух и три полученных числа выводятся в качестве ответа. Если же всё наоборот, то находится меньшее из первых двух чисел (x, y) и его значение заменяется на полусумму оставшихся двух числовых значений. На выходе выводятся также три числа, но одно из них будет заменено в соответствии с правилами, описанными выше. Например, при вводе чисел x = -5, y = 5, z = -10: общая сумма чисел равна -10, что меньше 1. Значит выполняется первое условие, и благодаря этому условию, следующим условием путём попарного сравнения чисел находится меньшее из трёх чисел, в данном случае z = -10. Далее вычисляется полусумма х и у, в конкретном случае равная 0, и начальное число z заменяется этой полусуммой. Благодаря таким преобразованиям на выходе получатся числа: -5, 5, 0.

**4. Описание алгоритма**

Начало

4.1 **Ввод исходных чисел x, y, z**

4.2 **Проверка условия, при котором данные числа недопустимы для ввода (равны между собой).** **Если условие выполняется, то программа прекращает свою работу, оповещая пользователя о том, что введённые им числа недопустимы для ввода**

4.3 **Производится процесс сложения трёх данных чисел**

4.4 **Проверка условия, при котором общая сумма чисел меньше 1.** **Если условие не выполняется, то происходит переход к пункту 4.12**

4.5 **Проверка условия, при котором х – наименьшее из трёх чисел (при общей сумме меньше 1).** Если условие выполняется, то вычисляется (z + у) / 2 и значением выражения заменяется число х

4.6 Выводятся три конечных числа и программа завершает работу

4.7 Проверка условия, при котором у - наименьшее из трёх чисел (при общей сумме меньше 1). Если условие не выполняется, то осуществляется переход к пункту 4.10

4.8 Вычисляется (х + z) / 2 и получившимся значением заменяется у

4.9 В качестве ответа выводятся три конечных числа и программа завершает свою работу

4.10 Вычисляется (х + у) / 2 и этим значением заменяется число z

4.11 В ответе выводятся также три получившихся числа и происходит завершение работы программы

4.12 Проверяется условие, при котором х – наименьшее из двух чисел х и у (при общей сумме чисел большей или равной 1). Если это условие выполняется, то вычисляется (y + z) / 2 и этим значением заменяется число х

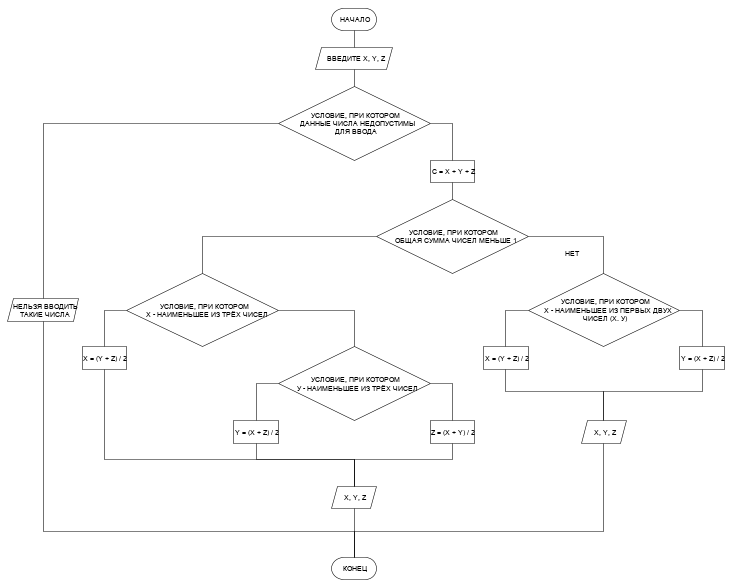
4.13 В ответе выводятся три получившихся числа и программа завершает работу

4.14 Вычисляется (x + z) / 2 и этим значением заменяется число у

4.15 В ответе выводится также три конечных числа и происходит завершение работы программы

Конец

**5. Опорный граф (блок-схема) алгоритма**



**6. Проект программы с определением замкнутых программных единиц и необходимых структур данных**

В данном случае программа должна состоять из трех частей – блок ввода, блок вывода и блок проверки условий, с выполнением математических операций, для вывода конечного ответа. В качестве структурных данных могут быть введены переменные для хранения исходных чисел (x, y, z).

**7. Текст программы:**

**на языке С++**

#include <iostream>;

#include <math.h>;

using namespace std;

int main()

{

float x, y, z;

setlocale(0, "");

cout << "Введите три попарно различных действительных числа: ";

cin >> x >> y >> z; //Ввод трёх попарно различных чисел

if ((x == y) || (x == z) || (y == z))

cout << "Нельзя вводить такие числа, введите другие числа" << endl; //Проверка условия их неравенства друг другу

else

{

if (x + y + z < 1) //Условие, при котором общая сумма чисел меньше 1

{

if ((x < y) && (x < z)) //Условие, при котором х - наименьшее из чисел

x = (y + z) / 2;

else if ((y < x) && (y < z)) //Условие, при котором у - наименьшее из чисел

y = (x + z) / 2;

else

z = (x + y) / 2;

cout << x << endl << y << endl << z << endl;

}

else

{

if (x < y) //Проверка условия, при котором х - наименьшее из первых двух чисел

x = (y + z) / 2;

else

y = (x + z) / 2;

cout << x << endl << y << endl << z << endl;

}

}

return 0;

}**на языке Pascal**

**Program** lab\_2\_2;

**Var** x, y, z: real;

**Begin**

Writeln ('Введите три попарно различных действительных числа');

Readln (x, y, z); //Ввод трёх попарно различных действительных чисел

**if** ((x = y) **or** (x = z) **or** (y = z)) **then**

Writeln ('Нельзя вводить такие числа введите другиие числа') //Проверка условия, при котором все числа, собственно, будут различны

**else**

**if** (x + y + z < 1) **then** //Условие, при котором сумма меньше 1

**begin**

**if** ((x < y) **and** (x < z)) **then** //Условие, при котором х - наименьшее из представленных чисел

x := (x + y) / 2

**else if** ((y < x) **and** (z < x)) **then** //Условие, при котором у - наименьшее из представленных чисел

y := (x + z) / 2

**else**

z := (x + y) / 2;

Writeln (x);

Writeln (y);

Writeln (z);

**end**

**else**

**begin**

**if** (x < y) **then** //Условие, при котором х - меньшее из первых двух чисел

x := (y + z) / 2

**else**

y := (x + z) / 2;

Writeln (x);

Writeln (y);

Writeln (z);

**end**;

**End**.**8. Проверка работоспособности (тестирование) программы**

Для проверки работоспособности по запросу (после запуска программы) введем числа: x = 1, y = 2, z = 3. На выходе должно получиться число x = 2,5, y = 2, z = 3.

**9. Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду своей простоты программа одинаково быстро выполняется на Pascal и C++.