1. **Цель работы**

Целью работы является изучение логических операций типа НЕ, И, ИЛИ.

1. **Задание**

Согласно варианту №3 необходимо определить, попадает ли точка в область:

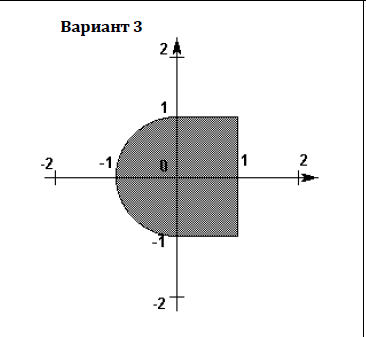


Рисунок 1 – Задания по варианту

1. **Описание созданных функций**

**Имя:** get\_num

**Назначение:** Запрос и проверка числа на корректность

**Входные данные:**

* -

**Выходные данные:**

* double x **–** проверренное число

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
| Вход | Выход |
| 5g | Неверный ввод |
| 5 | 5 |

**Прототип:** double get\_num()

**Псевдокод**

Запросить число

Проверить на корректность

Вернуть число

**Блок-схема**

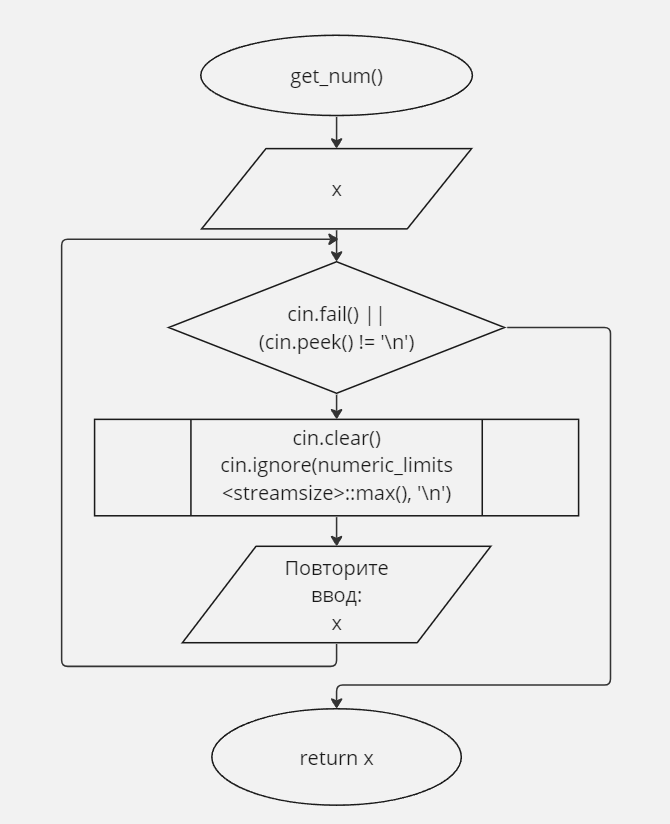


Рисунок 2 – Блок схема get\_num

**Имя:** request

**Назначение:** Запрос координаты

**Входные данные:**

* double y – Координата для последующего вывода через флаг

**Выходные данные:**

* double x – Координата x

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
|  | x, y |
| 0.5 0 | 0.5 0 |
| 5g 0 | Неверное значение |

**Прототип:** double request()

**Псевдокод**

Запросить координату x и y

Проверить на корректность

Вернуть x и y

**Блок-схема**

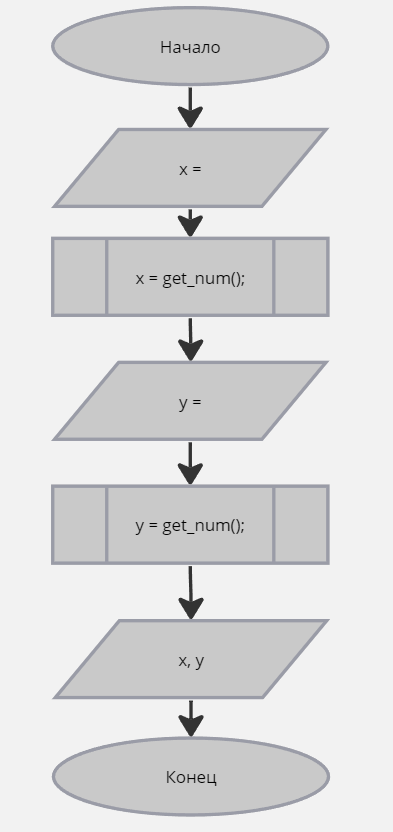


Рисунок 3 – Блок схема request

**Имя:** check\_right

**Назначение:** Проверяет, входит ли точка в правую часть системы координат

**Входные данные:**

* double x – Координата x
* double y – Координата y

**Выходные данные:**

* bool – Входит(true) или нет(false)

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |  |
| --- | --- | --- |
| x | y | Выход |
| 0.5 | 0.2 | true |
| 2 | 0 | false |

**Прототип:** bool check\_right(const double x, const double y)

**Псевдокод**

Получить координату x и y  
 Проверить на вхождение  
 Вернуть true или false

**Блок-схема**

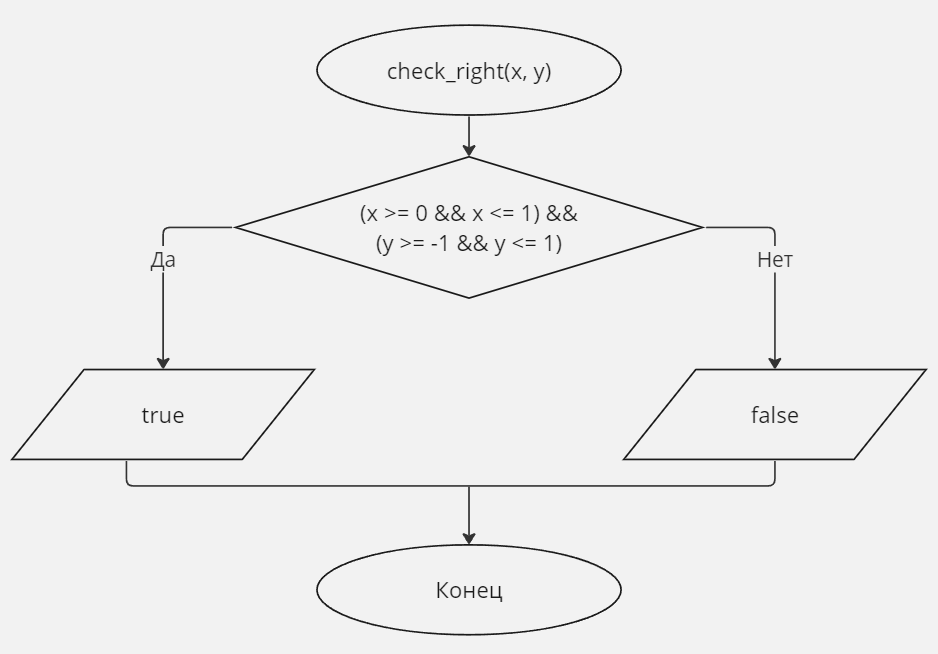


Рисунок 4 – Блок схема check\_right

**Имя:** check\_left

**Назначение:** Проверяет, входит ли точка в левую часть системы координат

**Входные данные:**

* double x – Координата x
* double y – Координата y

**Выходные данные:**

* bool – Входит(true) или нет(false)

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |  |
| --- | --- | --- |
| x | y | Выход |
| -0.5 | 0.5 | true |
| -10 | 0.5 | false |

**Прототип:** bool check\_left(const double x, const double y)

**Псевдокод**

Получить координату x и y  
 Проверить на вхождение  
 Вернуть true или false

**Блок-схема**

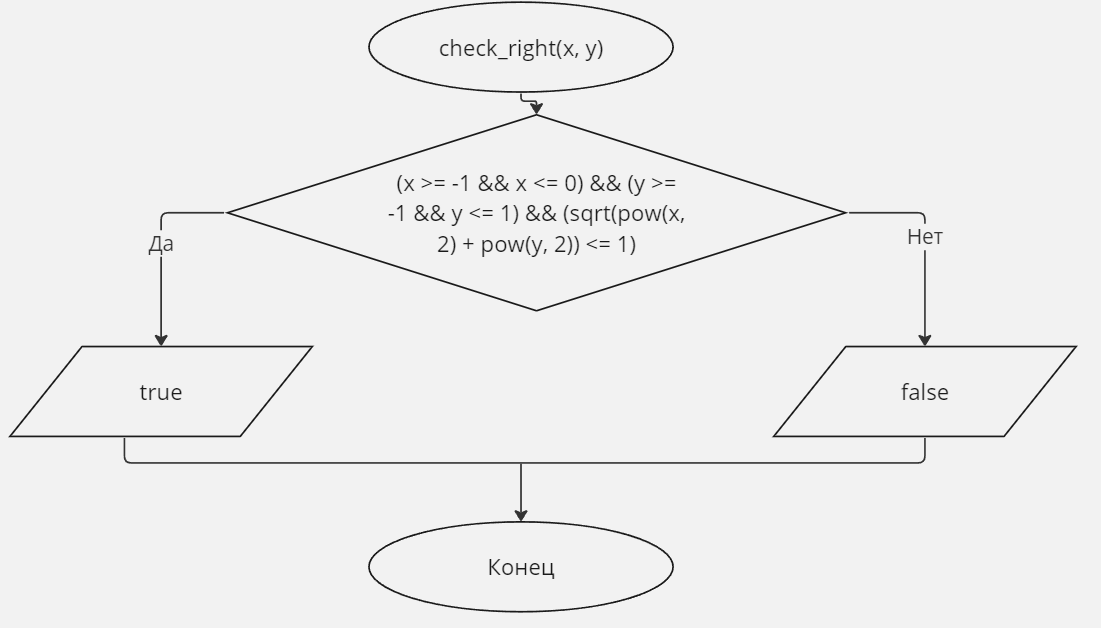


Рисунок 5 – Блок схема check\_left

1. **Текст программы**

#include<iostream>

#include <limits>

#include <math.h>

using namespace std;

double get\_num() // Запрос и проверка числа на корректность

{

double x;

cin >> x;

while (cin.fail() || (cin.peek() != '\n')) // Проверка на корректность

{

cin.clear(); // Очищение флага ошибки

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Очистка буфера запроса

cout << "Повторите ввод: ";

cin >> x;

}

return x;

}

double request(double &y) // Функция запроса координаты x

{

double x;

cout << "Введите координаты точек" << endl;

cout << "x = ";

x = get\_num();

cout << "y = ";

y = get\_num();

return x;

}

bool check\_right(const double x, const double y) // Функция проверки вхождения в 1 четверть

{

if ((x >= 0 && x <= 1) && (y >= -1 && y <= 1))

{

return true;

}

else

{

return false;

}

}

bool check\_left(const double x, const double y) // Функция проверки вхождения во 2 четверть

{

if ((x >= -1 && x <= 0) && (y >= -1 && y <= 1) && (sqrt(pow(x, 2) + pow(y, 2)) <= 1))

{

return true;

}

else

{

return false;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

double x, y;

cout << "Привет! Введи координаты точки!" << endl;

x = request(y);// Запрос координат

if (check\_right(x, y) || check\_left(x, y)) // Проверки на каждую четверть

{

cout << "Точка входит в область" << endl;

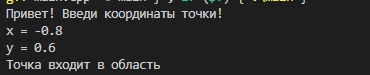
return 1;

}

cout << "Точка не входит в область" << endl;

}

1. **Пример работы программы**

Рисунок 6 – Пример работы программы

Полученные данные совпадают с действительными

1. **Анализ результатов и выводы**

В процессе лабораторной работы были изучены изучены логические операций типа НЕ, И, ИЛИ.

Достоинства программы:

* Некоторые пользовательские функции можно использовать в других работах.
* Данные проверяются на ввод во избежание ошибок.

Недостатки:

* Программа проверяет только ту область, что изначально задана.