



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА МЕДИЧНОЇ КІБЕРНЕТИКИ ТА ТЕЛЕМЕДИЦИНИ

Лабораторна робота № 4

з дисципліни «Алгоритмізація та програмування»
на тему: Алгоритми розгалуженої структури(умовний оператор)»

Варіант 16

Виконал:

Студент групи БС-32

Хоміщевич Микита

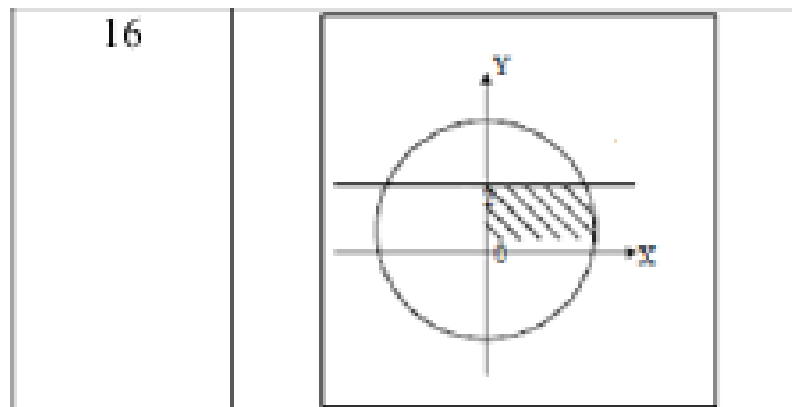
Перевірила:

викл.

Матвійчук А.О.

Задание

Відповідно до свого варіанту визначення ділянки на площині написати програму для встановлення належності заданої користувачем точки цій ділянці.



☐ Лабораторна робота виконана без зауважень

☐ Лабораторна робота має зауваження:

☐ присутні зауваження до блок-схем:

☐ виконані не за стандартом

☐ не відповідають коду

☐ присутній код

☐ інші зауваження:

☐ присутні зауваження до коду:

☐ задача завдання вирішена хибно

☐ код програми не компілюється

☐ використано глобальні змінні

☐ недостатня декомпозиція на функціонуванні користувача

☐ статичні змінні при роботі з масивами

☐ оформлення коду

☐ присутні зайві символи «{» та «}»

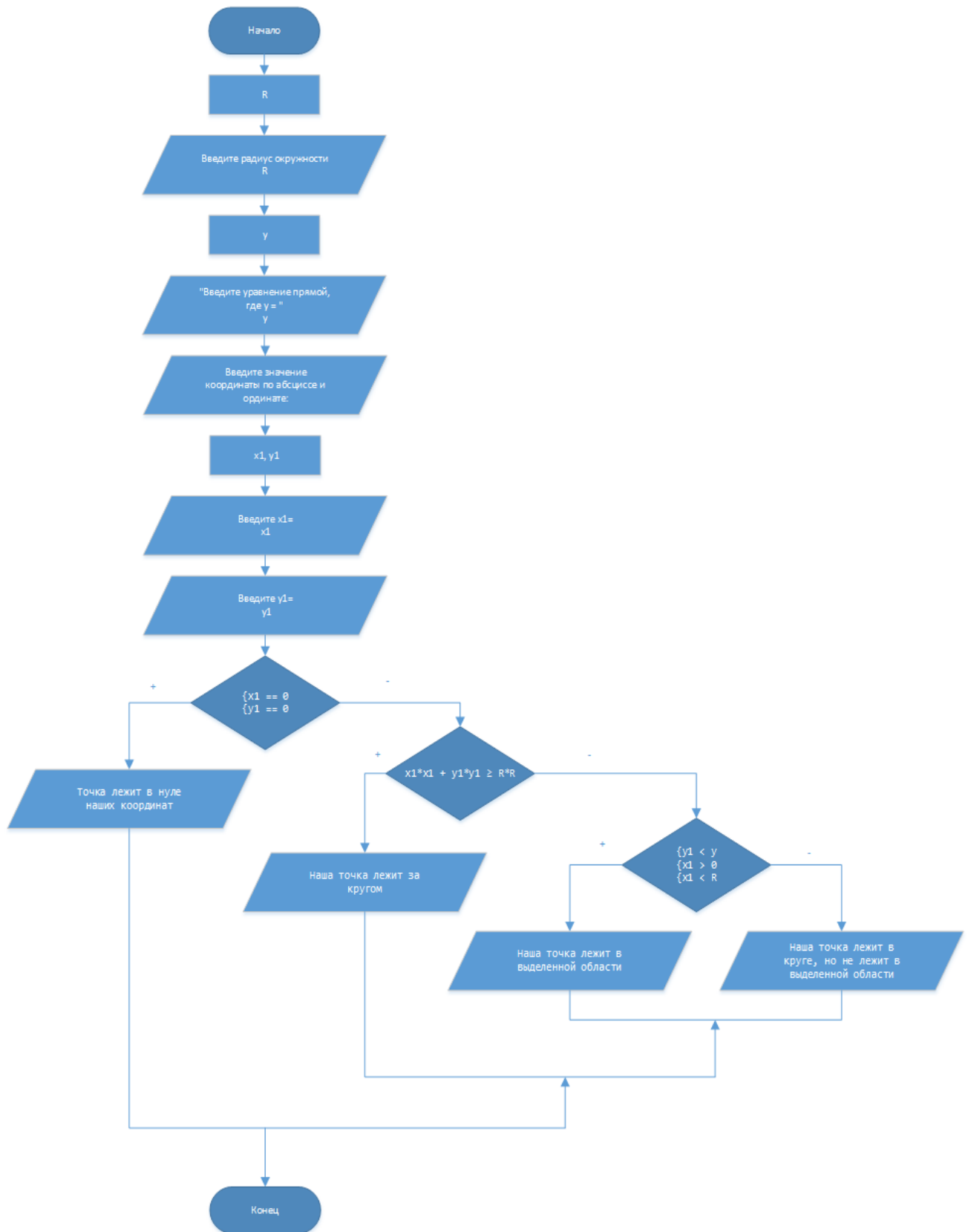
☐ інші зауваження:

☐ невірні відповіді на запитання:

☐ №1 ☐ №2 ☐ №3

☐ маються інші зауваження:

Блок – схема



Программный код

```
#include <iostream>

#include <conio.h>

#include <locale.h>

using namespace std;

int main() {

    setlocale(LC_ALL, "RUS");

    float R;

    cout << "Введите радиус окружности: ";

    cin >> R;

    float y;

    cout << "Введите уравнение прямой, где y = ";

    cin >> y;

    cout << "Введите значение координаты по абсцисе и по ординате: " << endl;

    float x1, y1;

    cout << "Введите x1=";

    cin >> x1;

    cout << "Введите y1=";

    cin >> y1;

    if (x1 == 0 && y1 == 0)

    {

        cout << "Точка лежит в нуле наших координат" << endl;

    }

    else

    {

        if (x1*x1 + y1*y1 >= R*R)

        {

            cout << "Наша точка лежит за кругом" << endl;

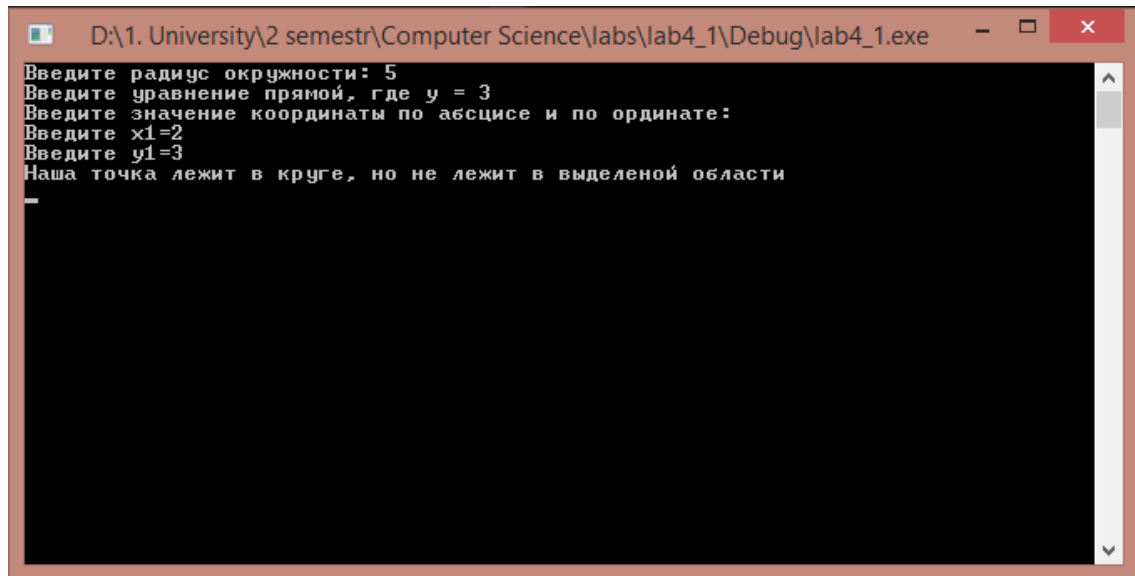
        }

    }

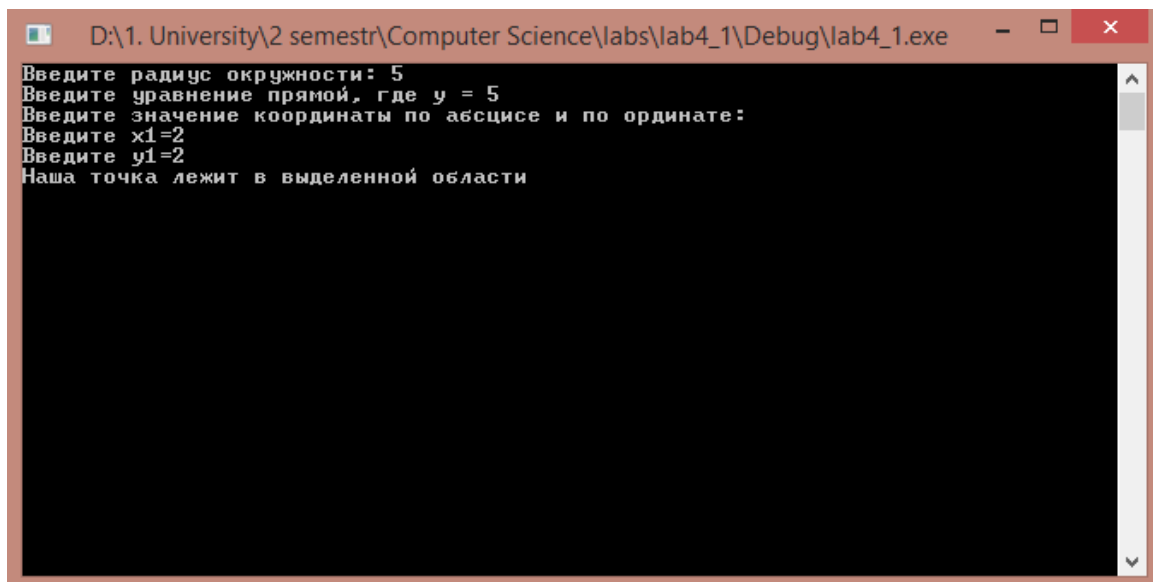
}
```

```
        else
        {
            if (y1 < y && x1 > 0 && x1 < R)
            {
                cout << "Наша точка лежит в выделенной области" << endl;
            }
            else
            {
                cout << "Наша точка лежит в круге, но не лежит в выделенной области" <<
endl;
            }
        }
    }
    _getch();
}
```

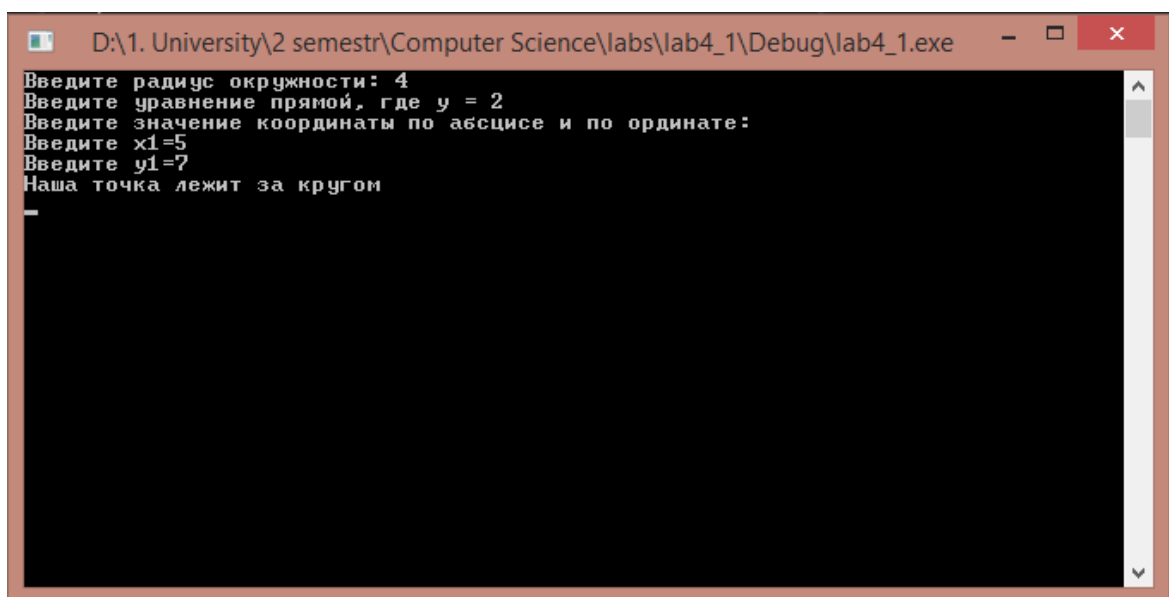
Скриншоты работы программы



```
D:\1. University\2 semestr\Computer Science\labs\lab4_1\Debug\lab4_1.exe
Введите радиус окружности: 5
Введите уравнение прямой, где  $y = 3$ 
Введите значение координаты по абсциссе и по ординате:
Введите  $x1=2$ 
Введите  $y1=3$ 
Наша точка лежит в круге, но не лежит в выделенной области
```



```
D:\1. University\2 semestr\Computer Science\labs\lab4_1\Debug\lab4_1.exe
Введите радиус окружности: 5
Введите уравнение прямой, где  $y = 5$ 
Введите значение координаты по абсциссе и по ординате:
Введите  $x1=2$ 
Введите  $y1=2$ 
Наша точка лежит в выделенной области
```



```
D:\1. University\2 semestr\Computer Science\labs\lab4_1\Debug\lab4_1.exe
Введите радиус окружности: 4
Введите уравнение прямой, где  $y = 2$ 
Введите значение координаты по абсциссе и по ординате:
Введите  $x1=5$ 
Введите  $y1=7$ 
Наша точка лежит за кругом
```

Ответы на контрольные вопросы

1. Что такое алгоритм разветвленной структуры, программа разветвленной структуры?

Алгоритм розгалуженої структури – це алгоритм, в якому розрахунки виконуються за однією з гілок.

2. В чем состоит разница между операциями и операторами?

Оператор – это команда или последовательность инструкций, которая что-то исполняет. Пример, оператор ветвления: `if`

Операция – это способ задания некоторых действий. В данном случае, это действие, которое исполняется (с какими-то данными или при их отсутствии). Пример: `if (a > b) {c = a;}`

3. Определите синтаксис условного оператора

Умовний оператор має скорочену (без *else*) або повну (з *else*) форми. За допомогою скороченої форми умовного оператора можливо виконати або ні оператори, що розташовані після умови, в залежності від істинності або хибності умови, що аналізується. Формат скороченої форми умовного оператора:

```
if ( expression )
    statement;
```

де *expression* – умова прийняття рішення, *statement* – простий або складений оператор.

Повна форма умовного оператора дозволяє вибрати для виконання один з двох можливих операторів, що відповідають одній з двох можливих альтернатив процесу обрахування. Формат повної форми умовного оператора:

```
if ( expression )
    statement1;
else
    statement2;
```

де *expression* – умова прийняття рішення, *statement1* та *statement2* – простий або складений оператор.

4. Разница условного оператора с одной веткой и двумя ветками

При скорашенном условии (одна ветка) выполнится только единственно заданное условие, если на входе будет условие, что не удовлетворяет данную ветку, то ничего не выполнится, т.к. нет еще одного условия.

При условном операторе с двумя ветками (как правило, `if() {...}; else {...}`) если наше условие = `false`, тогда выполняется другое условие, записанное в `else`.

5. Что такое логическое выражение? Перечислите логические операции, которые выполняются в логических выражениях при написании программ на языке C++

Логическое выражение — конструкция языка программирования, результатом вычисления которой является «истина» или «ложь».

Логические операции:

- Логическое «И», синтаксис: &&
- Логическое «ИЛИ», синтаксис: ||
- Логическое «НЕ», синтаксис: !

6. Как строится сложное условие, которое объединяет несколько условий?

Сложное условие – условие в языке программирования С, которое состоит из нескольких простых условий, объединенных логическими операциями.

Пример:

```
if ((a > b) && (b > c)) {x = a;}
```

Здесь мы имеем сложное условие, которое состоит из двух простых (строгое сранвление данных), объединенное логической операцией «И», так как нам надо для верного результата иметь значение true в двух условиях.