 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №4**

з дисципліни «Алгоритмізація та програмування»

на тему: «Алгоритми розгалуженої структури (умовний оператор)»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

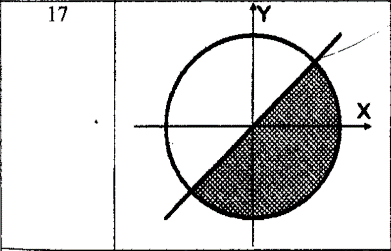
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2019

**Завдання:**

1. Вивчити теоретичні основи написання алгоритмів розгалуженої структури з використанням умовного оператора.
2. Відповідно до свого варіанту розробити математичну модель для визначення ділянки площини: навести рівняння ліній, що обмежують заштриховані ділянки, та описати умови попадання точки в кожну ділянку.



Рівняння ліній:

Умови попадання точки в ділянку:

1. Побудувати блок-схему алгоритму вирішення завдання.
2. Відповідно до свого варіанту та розробленої математичної моделі розбиття площини написати програму для визначення умов попадання заданої користувачем точки в кожну з ділянок.
3. Скласти і захистити звіт по роботі.

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок-схема:**



**Код:**

#include <iostream>

#include <math.h>

#include <conio.h>

int main()

{

float x = 0, y = 0, R = 0; // ввод переменных

std::cout << "Enter x:\n";

std::cin >> x;

std::cout << "Enter y:\n";

std::cin >> y;

std::cout << "Enter radius R:\n";

std::cin >> R;

if ((x \* x + y \* y) > (R \* R)) // за пределами круга

std::cout << "The point is outside the circle\n";

else

{

std::cout << "The point is inside the circle ";

if (x < y) // над прямой y=x

std::cout << "and above the line";

else // в нужной области

std::cout << "and under the line\nSo the point is in the area";

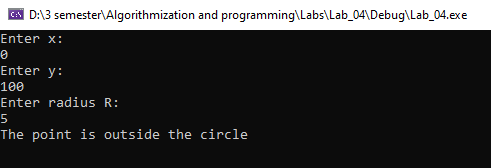
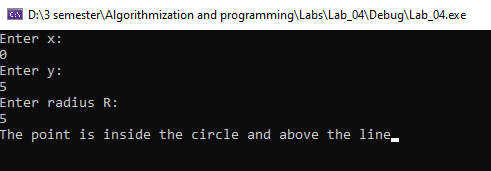
}

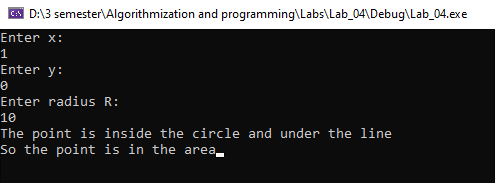
\_getch();

return 0;

}

**Результати:**

****



**Контрольні питання:**

1. ***Що таке алгоритм розгалуженої структури?***

Алгоритм розгалуженої структури – це алгоритм, в якому деякі дії виконуються(або не виконуються) в залежності від поставлених умов.

1. ***Визначити синтаксис умовного 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 )

statement 1;

else:

statement 2;

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

1. ***Яка різниця між умовними операторами з однією та двома гілками?***

Умовний оператор з однією гілкою може виконати певні оператори лише за вірною умовою прийняття рішення.

Умовний оператор з двома гілками, має альтернативу у виді else, яка виконається навіть у разі False в умові. Таким чином, умовний оператор з двома гілками у перспективі може виконати 2 оператори при True та False, коли умовний оператор з однією гілкою може лише виконати чи не виконати певний оператор при різних бульових значеннях.

1. ***Чи можливе існування тільки гілки else (без операторів if-умови) в умовному операторі?***

Неможливе, бо якщо немає умови, то не існує і варіантів розвитку подій.

1. ***Коли в умовному операторі слід використовувати фігурні дужки «{» та «}», а коли ні?***

Коли в операторі більше одної дії, тоді ми використовуємо фігурні дужки, а коли одна дія то їх можна не використовувати.

1. ***Що таке логічний вираз?***

Логічний вираз - це конструкція мови програмування, результат обчислення якої може бути істинним (тобто значення логічного виразу дорівнює true або будь-якому цілочисельному значенню, окрім «0») або хибним (тобто значення логічного виразу дорівнює false або «0»).

1. ***Перелічіть логічні операції, що використовуються в логічних виразах при написанні програм мовою С++.***

У с++ існують три логічні операції:

Логічна операція «і» - &&

Логічна операція «або» - ||

Логічна операція «НЕ» або логічне заперечення - !

1. ***Яка операція («&&» чи «||») має більший пріоритет? Навести відповідний приклад.***

Логічна операція «і» - && має більший пріоритет ніж «або» - ||

val = a || b && c;

Спочатку виконається логічна операція «і», потім – «або». Через більшу пріорітетність першої операції.

1. ***Як будується складна умова, що об’єднує декілька умов?***

Складна умова об’єднується за допомогою логічних операцій.

1. ***Чи виконається в наступному коді одна з гілок умовного оператора?***

int x=0, y=-1;

if ( x=0 )

y=x;

else

y=1;

cout « y;

Умова рівності перевіряється двома знаками «=» (e.g. x == 0). Виконується гілка else, бо результатом присвоєння є присвоєний результат, тобто «0» - логічний False. На виводі маємо «1».