МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Технічна Кібернетика»

**Лабораторна робота №1**

**з дисципліни «Основи программування»**

**на тему: «**Розробка та виконання програми лінійної структури на мові С++. Середовище розробки та виконання програм Microsoft Visual Studio.**»**

Виконав:

студент гр.ПЗ1911

Сафонов Д.Є.

Прийняла:

Нежуміра О.И.

Дніпро, 2019

**Тема.** Розробка програм з розгалуженнями.

**Мета.** Сформувати навички розробки алгоритмів розгалуженої структури і реалізації їх на мові С++. Навчитися конструювати складені умови за допомогою логічних операцій.

***Задачи:***

**Программа 1:** Дано дійсні додатні числа a, b, c, d. З`ясувати, чи можна прямокутник зі сторонами a, b вмістити всередині прямокутника зі сторонами c, d так, щоб кожна зі сторін одного прямокутника була паралельна або перпендикулярна кожній стороні другого прямокутника.

***1. Зовнішні специфікації;***

*1.1.Формат вхідних даних*

Таблиця 1

|  |  |  |  |
| --- | --- | --- | --- |
| Найменування даних | Умовне позначення | Вимоги до данних | Приклад |
| Число a | a | Дійсне число більше за нуль | 1 |
| Число b | b | Дійсне число більше за нуль | 2 |
| Число c | c | Дійсне число більше за нуль | 3 |
| Число d | d | Дійсне число більше за нуль | 4 |

*1.2.Формат вихідних даних*

Строка true/false

***2. Метод рішення завдання;***

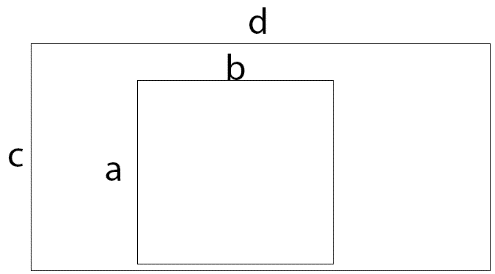


Рисунок 1

Якщо усі сторони більше чім нуль та обидві сторони внутрішнього прямокутника менше ніж сторони зовнішнього, то відповідь – так, якщо усе хоч одне з тверджень не є правдою, то ні.

a,b,c,d>0

c>a та d>b; або c>b та d>a;

***3. Тести для перевірки правильності програми;***

Таблиця 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назва | Вхідні дані | | | | Очікувані результати |
|  | a | b | c | d |  |
| Корректні дані | 1 | 2 | 3 | 4 | True |
| Граничне a | 0 | 1 | 1 | 1 | False |
| Граничне b | 1 | 0 | 1 | 1 | False |
| Граничне c | 1 | 1 | 0 | 1 | False |
| Граничне d | 1 | 1 | 1 | 0 | False |
| Нульові | 0 | 0 | 0 | 0 | False |
| Некорректне a | -1 | 2 | 3 | 4 | False |
| Некорректне b | 1 | -2 | 3 | 4 | False |
| Некорректне c | 1 | 2 | -3 | 4 | False |
| Некорректне d | 1 | 2 | 3 | -4 | False |

***4. Алгоритм розв’язання задачі***

*4.1. Блок схема*

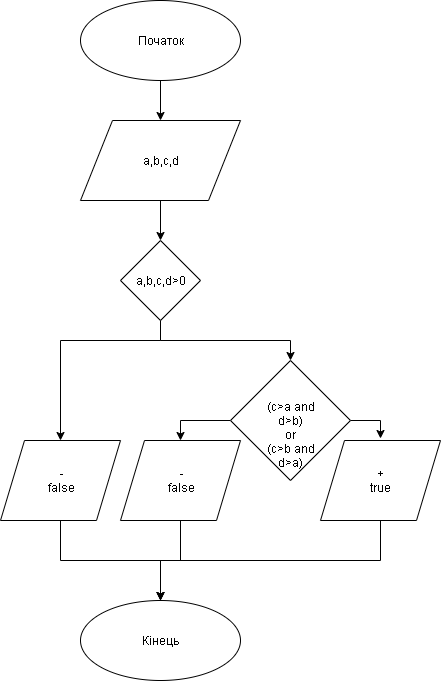


Рисунок 2

*4.2. Діаграмма Нассі-Шнейдермана*

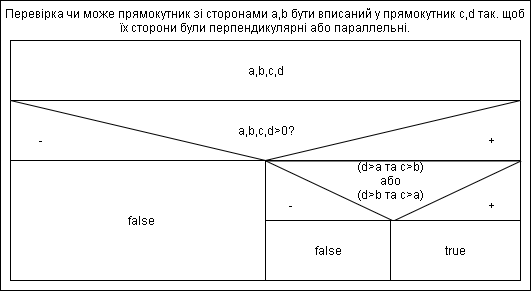


Рисунок 3

***5. Текст програми;***

#include <iostream>

#include <Windows.h>

using namespace std;

//Body

int main()

{

double a, b, c, d;//initialising variables

cout << "Programm checks if rectangle with sides of a,b can be placed in rectangle with sides of c,d" << endl << "so that their sides parallel/perpendicular" << endl;

cout << "input a ";//input

cin >> a;

cout << "input b ";

cin >> b;

cout << "input c ";

cin >> c;

cout << "input d ";

cin >> d;

if (a > 0 && b > 0 && c > 0 && d > 0) {//checks if all numbers >0

if ((a<c && b<d) || (a<d && b<c)) {//checks if first rectangle has both sides bigger than second

cout << "true" << endl;

}

else {

cout << "false" << endl;

}

}

else {

cout << "one or more numbers <0; false" << endl;

}

system("pause");

return 0;

}

***6. Результати тестування програми та їх аналіз;***

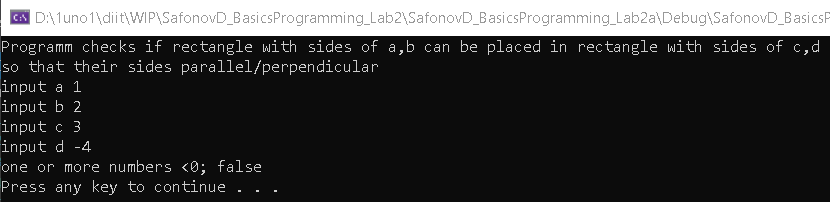


Рисунок 4

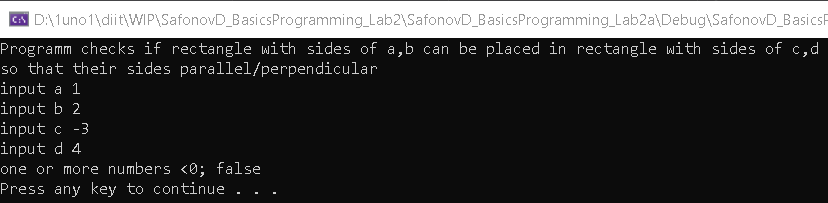


Рисунок 5

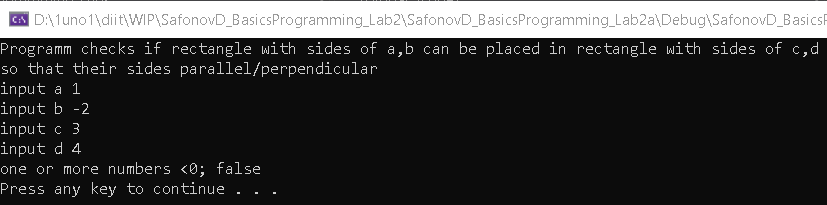


Рисунок 6

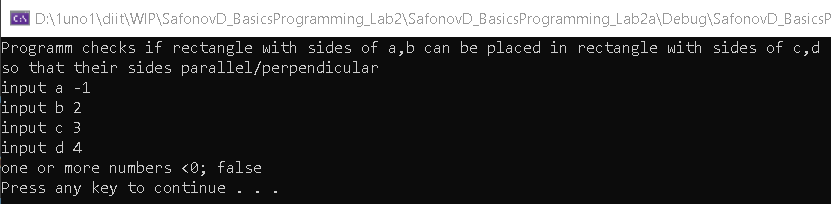


Рисунок 7

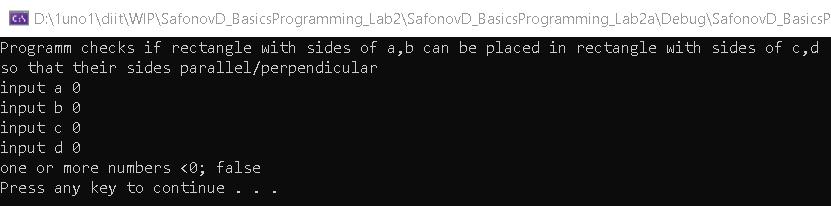


Рисунок 8

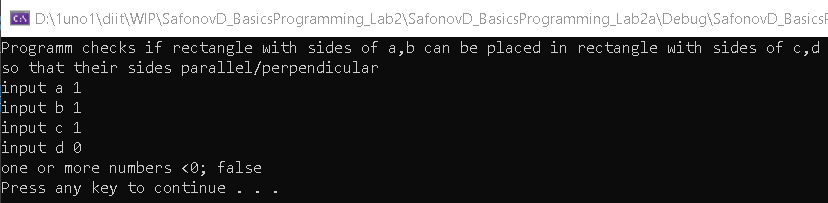


Рисунок 9

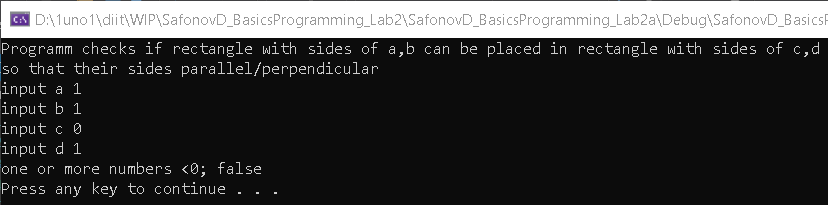


Рисунок 10

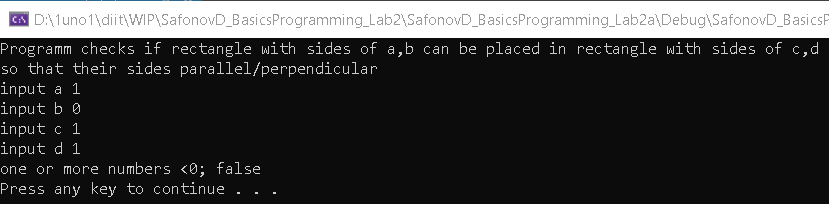


Рисунок 11

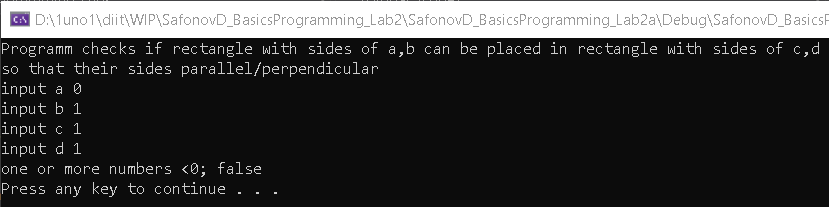


Рисунок 12

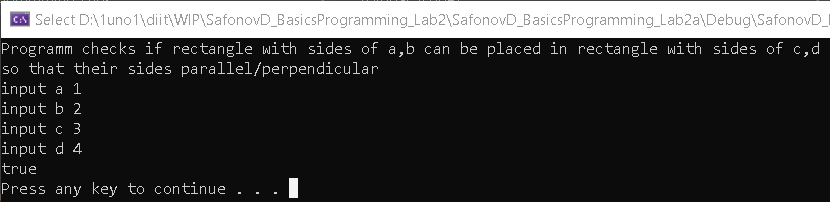


Рисунок 13

Відповіді наданні программою відповідають очікуванним.

***7. Висновки щодо призначення та використання операторів розгалужень та логічних операції.***

&& - та; || - або; якщо потрібно використати складне розгалуження, то треба використовувати дужки, щоб відокремити прості розгалуження та комбінувати їх між собою.

**Программа 2:** Трикутник задано координатами своїх кінців. Знайти координати прямокутника, в який можна помістити заданий трикутник. Різниця площ фігур повинна бути мінімальною.

***1. Зовнішні специфікації;***

*1.1. Формат вхідних даних*

Таблиця 3

|  |  |  |  |
| --- | --- | --- | --- |
| Найменування даних | Умовне позначення | Вимоги до данних | Приклад |
| Координата x точки a | Xa | Дійсне число | 0 |
| Координата y точки a | Ya | Дійсне число | 0 |
| Координата x точки e | Xe | Дійсне число | 1 |
| Координата y точки e | Ye | Дійсне число | 1 |
| Координата x точки d | Xd | Дійсне число | 2 |
| Координата y точки d | Yd | Дійсне число | 0 |

*1.2. Формат вихідних даних*

Таблиця 4

|  |  |  |  |
| --- | --- | --- | --- |
| Найменування даних | Умовне позначення | Вимоги до данних | Приклад |
| Координата x точки a | Xa | Дійсне число | 0 |
| Координата y точки a | Ya | Дійсне число | 0 |
| Координата x точки b | Xb | Дійсне число | 0 |
| Координата y точки b | Yb | Дійсне число | 1 |
| Координата x точки c | Xc | Дійсне число | 2 |
| Координата y точки c | Yc | Дійсне число | 1 |
| Координата x точки d | Xd | Дійсне число | 2 |
| Координата y точки d | Yd | Дійсне число | 0 |

***2. Метод рішення завдання;***

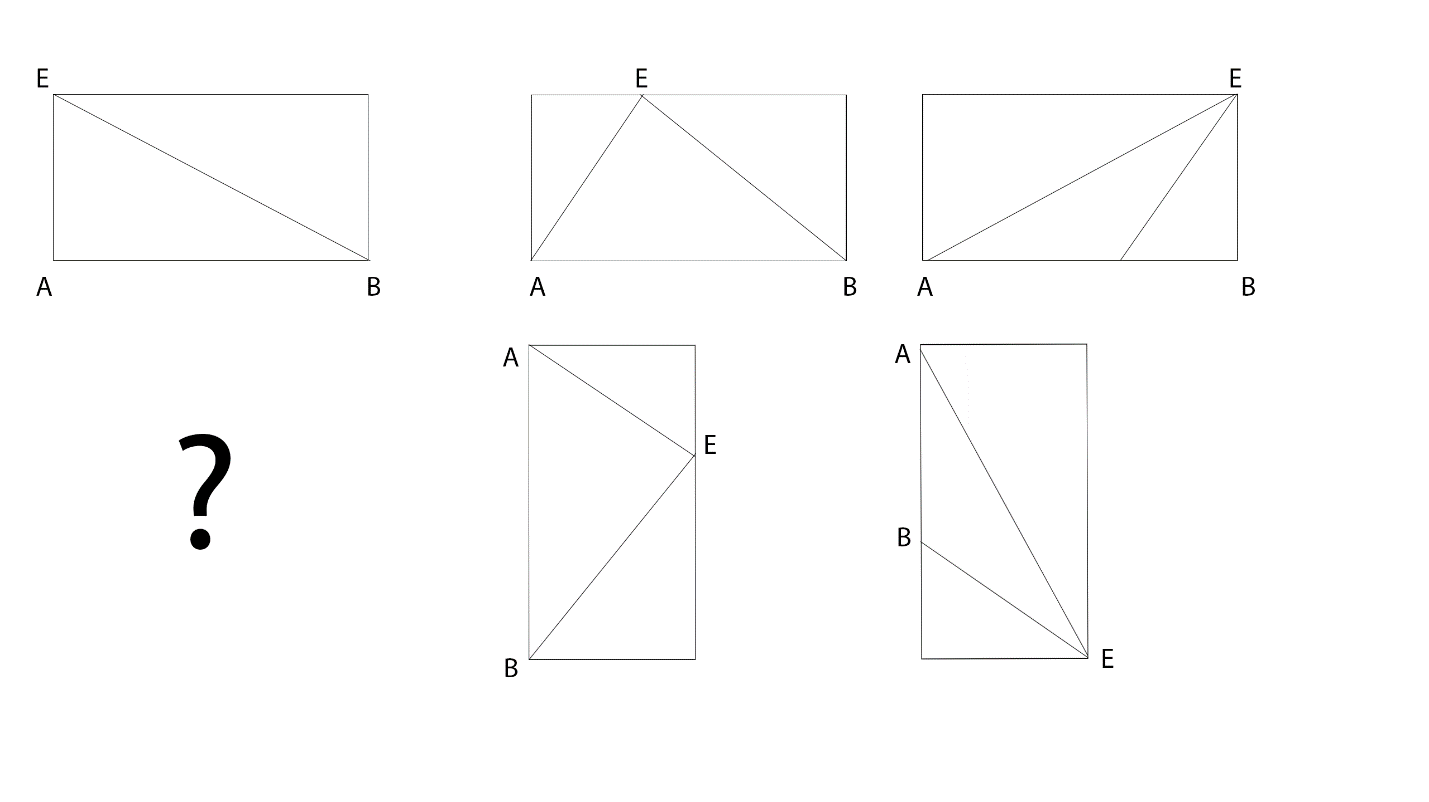


Рисунок 14

Існує декілька основних випадків, на які можна розділити це завдання:

1)Довільний трикутник, який не підпадає під інші випадки – випадок вирішується тільки за загальною формулою.

2)Прямокутний трикутник, катети лежать на вертикалі/горізонталі.

3)AB лежить вертикально, E нижче/више та за иксом поміж A та B.

4)випадок 3 повернутий на 90 градусів.

5) AB лежить горизонтально, E правіше/лівіше та за игреком поміж A та B

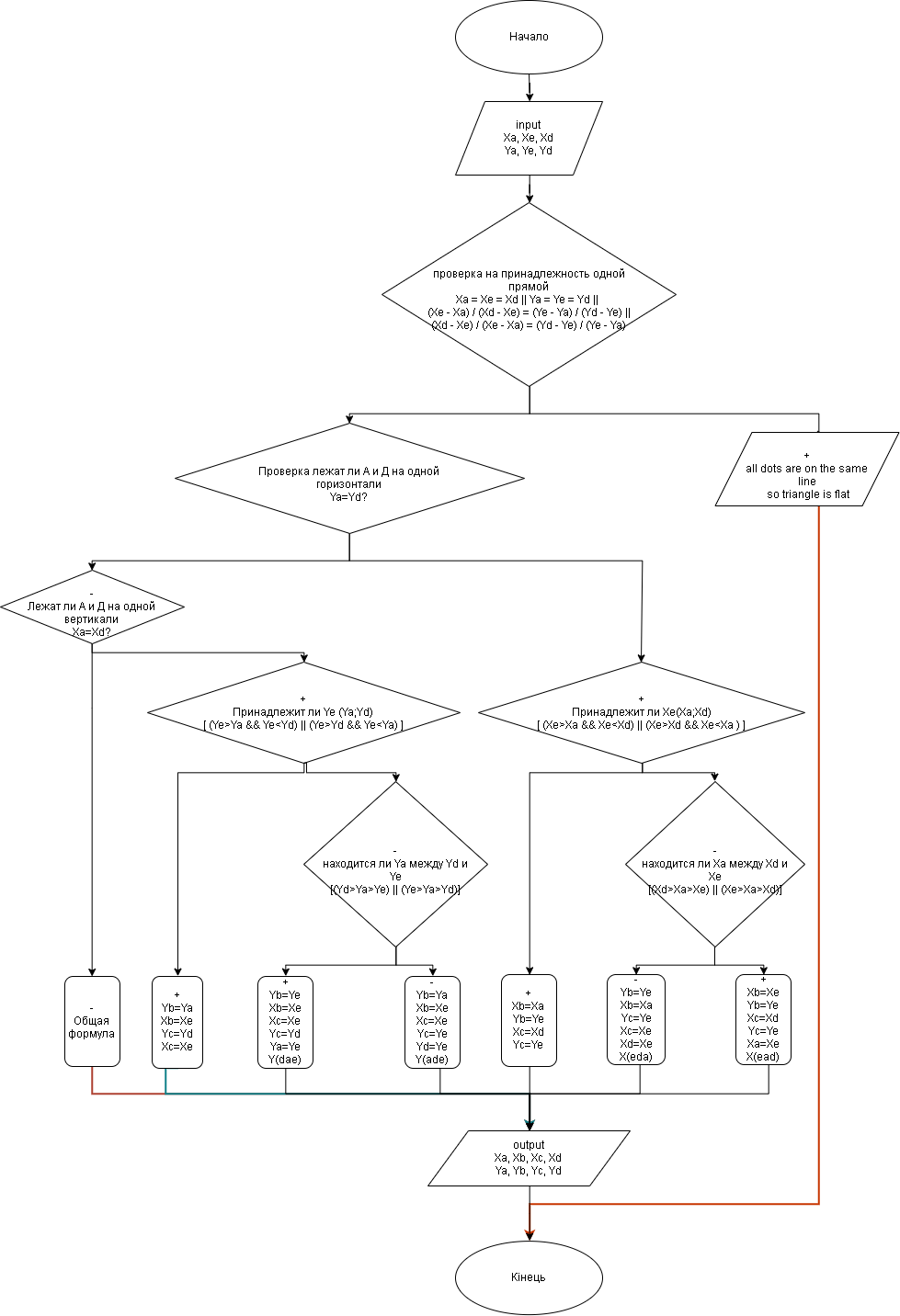
6)випадок 5 повернутий на 90 градусів.

***3. Тести для перевірки правильності програми;***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назва | Вхідні дані | | | Очікувані результаті | |
| (x;y) | A | E | D | B | C |
| 1 | (2;1) | (3;4) | (4;3) | (0;2) | (3;4) |
| 2 | (0;0) | (1;0) | (0;1) | (0;1) | (1;1) |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |

***4. Алгоритм розв’язання задачі***

*4.1. Блок-схема*



*4.2. Діаграма Нассі-Шнейдермана*

***5. Текст програми;***

//need to add uniformula

#include <iostream>

#include <Windows.h>

using namespace std;

//Body

int main()

{

double Xa, Ya, Xd, Yd, Xe, Ye;

#pragma region i\_o

cout << "Programm finds coordinates of rectangle vertexes" << endl << "so that triangle of given vertex coordinates is inscribed in rectangle" << endl << "and difference of their areas is as small as possible" << endl;

cout << "Triangle Vertexes(AED), Rectangle Vertexes(ABCD)" << endl;

cout << "input Xa ";

cin >> Xa;

cout << "input Ya ";

cin >> Ya;

cout << "input Xe ";

cin >> Xe;

cout << "input Ye ";

cin >> Ye;

cout << "input Xd ";

cin >> Xd;

cout << "input Yd ";

cin >> Yd;//in

#pragma endregion

double Xb, Yb, Xc, Yc;

if (((Xa == Xe) && (Xe == Xd)) || ((Ya == Ye) && (Ye == Yd)) || (((Xe - Xa) / (Xd - Xe)) == ((Ye - Ya) / (Yd - Ye))) || (((Xd - Xe) / (Xe - Xa)) == ((Yd - Ye) / (Ye - Ya)))) {

cout << "all 3 lines are in the same line, wrong input" << endl;//checking if all three dots are in the same line

system("pause");

return 0;

}

else {

if (Ya == Yd) {//checking if A and D are in the same horizontal

if (((Xe >= Xd) && (Xe <= Xa)) || ((Xe <= Xd) && (Xe >= Xa))) {//checking if Xe is between Xa and Xd regardless of which Xa or Xd is bigger

Xb = Xa;

Yb = Ye;

Xc = Xd;

Yc = Ye;

}

else {

if (((Xa >= Xe) && (Xa <= Xd)) || ((Xa <= Xe) && (Xa >= Xd))) {//checking if Xa inbetween Xd and Xe

Xb = Xe;

Yb = Ye;

Xc = Xd;

Yc = Ye;

Xa = Xe;

}

else {

Xb = Xa;

Yb = Ye;

Yc = Ye;

Xc = Xe;

Xd = Xe;

}

}

}

else {

if (Xa == Xd) {//checking if A and D are in the same vertical

if (((Ye >= Yd) && (Ye <= Ya)) || ((Ye <= Yd) && (Ye >= Ya))) {//checking if Ye is between Ya and Yd regardless of which Ya or Yd is bigger

Yb = Ya;

Xb = Xe;

Yc = Yd;

Xc = Xe;

}

else {

if (((Ya >= Ye) && (Ya <= Yd)) || ((Ya <= Ye) && (Ya >= Yd))) {//checking if Ya inbetween Yd and Ye

Yb = Ye;

Xb = Xe;

Yc = Yd;

Xc = Xe;

Ya = Ye;

}

else {

Yb = Ya;

Xb = Xe;

Xc = Xe;

Yc = Ye;

Yd = Ye;

}

}

}

else {

//add uniformula

cout << "uniformula isn't added yet";

system("pause");

return 0;

}

}

}

#pragma region rez\_out

//out

cout << "A(" << Xa << ";" << Ya << ")" << endl;

cout << "B(" << Xb << ";" << Yb << ")" << endl;

cout << "C(" << Xc << ";" << Yc << ")" << endl;

cout << "D(" << Xd << ";" << Yd << ")" << endl;

#pragma endregion

system("pause");

return 0;

}

***6. Результати тестування програми та їх аналіз;***

***7. Висновки щодо призначення та використання операторів розгалужень та логічних операції.***

&& - та; || - або; якщо потрібно використати складне розгалуження, то треба використовувати дужки, щоб відокремити прості розгалуження та комбінувати їх між собою.