МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Основи програмування»**

**на тему: «Розробка програм з розгалуженнями»**

Виконав: студент гр. ПЗ1911

Сіньков Г. О.

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

**Лабораторна робота 2**

**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.

Основи програмування

1.Постановка задачі.

**8)** Чи можна на прямокутної ділянці забудови розміром a на b метрів розмістити два будинки розміром в плані p на q і r на s метрів? Будинки можна розташовувати тільки паралельно сторонам ділянки.

2. Зовнішні специфікації

2.1.Вхідні дані

Вхідні дані: ширина та довжина ділянки, сторони першого будинку та сторони другого будинку.

**Формат вхідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Ширина ділянки | a | Ціле число | 10 |
| 2 | Довжина ділянки | b | Ціле число | 45 |
| 3 | Ширина буд. 1 | p | Ціле число | 6 |
| 4 | Довжина буд.1 | q | Ціле число | 7 |
| 5 | Ширина буд. 2 | r | Ціле число | 9 |
| 6 | Довжина буд.2 | s | Ціле число | 2 |

2.2. Формат вихідних даних

Вихідні дані: площа ділянки, площа першого будинку та площа другого будинку.

**Формат вихідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Площа ділянки | S | Ціле число | 450 |
| 2 | Площа буд. 1 | s1 | Ціле число | 42 |
| 3 | Площа буд. 2 | s2 | Ціле число | 18 |
| 4 | Можна поставити будинки на цій ділянці | - | - | - |
| 5 | Не можна поставити будинки на цій ділянці | - | - | - |
| 6 | Будинки не можуть перетинатись один з одним | - | - | - |
| 7 | Довжина сторона будинка перевищує довжина сторони ділянки | - | - | - |

2.3.Функціональні вимоги до програми

Програма повинна реалізовувати такі дії:

–введення даних у програму

–перевірку заданих даних

–вивід результату перевірки.

3. Вибір методу рішення задачі.

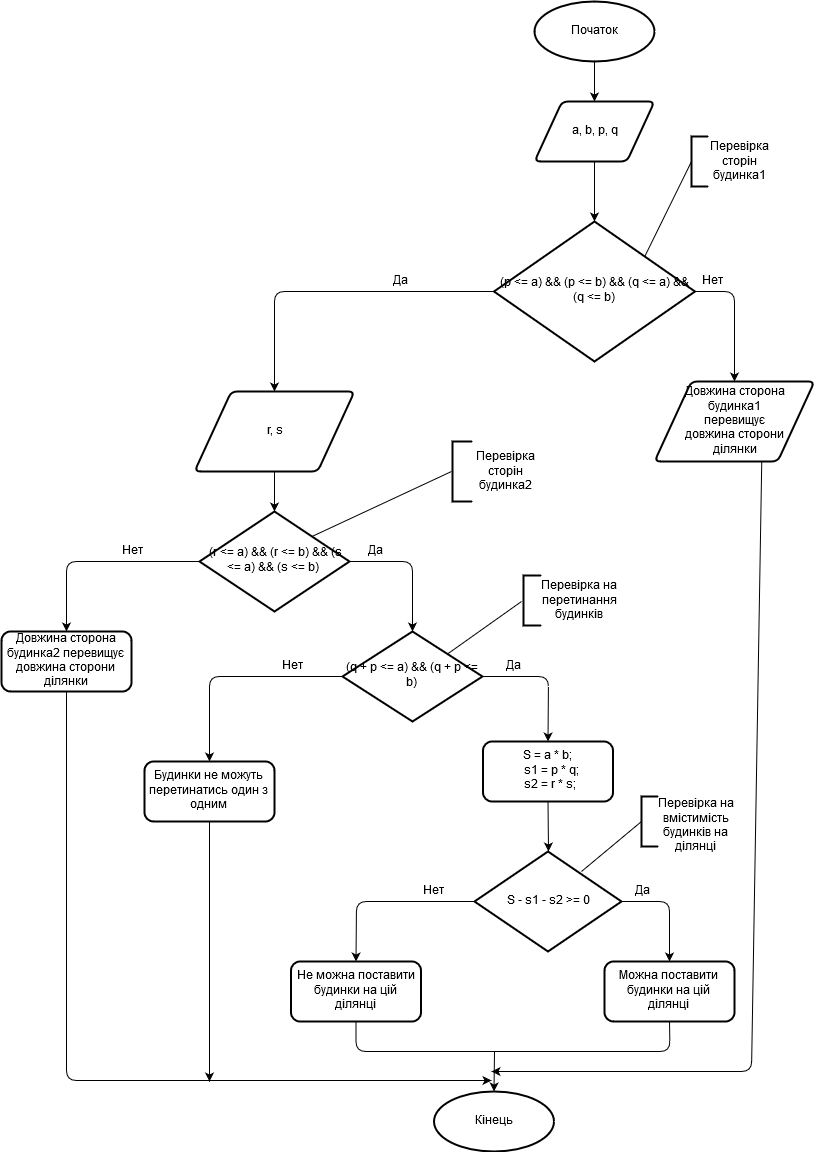
Перевірити на сумісність сторони будинка1 зі сторонами ділянки.( (p <= a), (p <= b), (q <= a), (q <= b), (r <= a), (r <= b), (s <= a), (s <= b) )

Перевірити на сумісність сторони будинка2 зі сторонами ділянки( (r <= a), (r <= b), (s <= a), (s <= b) )

Перевірка на перетинання будинків( S - s1 - s2 >= 0 )

Також здійснити перевірку площі ділянки та площі будинків (S - s1 - s2 >= 0)

4.Розробка алгоритму програми



5. Розробка тестів

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1 | Будинки можна розмістити | a=10 b=10 p=2 q=2 r=4 s=6 | S=100 s1=4 s2=24 |
| 2 | Сторона ділянки не може бути менше 0 | a=0 b=10 p=2 q=2 r=4 s=6 | S=0 s1=4 s2=24 |
| 3 | Сторони буд.1 не може бути більше за сторону будинка(як одна так і дві) | a=10 b=10 p=23 q=48 r=4 s=6 | S=100 s1=1104 s2=24 |
| 4 | Сторони буд.2 не може бути більше за сторону будинка(як одна так і дві) | a=10 b=10 p=2 q=2 r=78 s=61 | S=100 s1=4 s2=4758 |
| 5 | Максимальне значення сторін буд.1 | a=10 b=10 p=10 q=2 r=4 s=6 | S=100 s1=20 s2=24 |
| 6 | Максимальне значення сторін буд.2 | a=10 b=10 p=2 q=2 r=10 s=6 | S=100 s1=4 s2=60 |
| 7 | (Граничний по довжині буд. 2) Не можуть перетинатися | a=10 b=10 p=2 q=9 r=4 s=10 | S=100 s1=18 s2=40 |

6. Текст програми

/\*

Реалізувати введеня даних та всі можливі провірки сторон будинків да сторін ділянки.

\*/

#include<iostream>

#include<Windows.h>

using namespace std;

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, b, p, q, r, s;

int S, s1, s2;//S(ділянка), s1(буд.1), s2(буд.2)

cout << "Введіть сторону ділянки a: ";

cin >> a;

cout << "Введіть сторону ділянки b: ";

cin >> b;

cout << "Введіть сторону першого будинку p: ";

cin >> p;

cout << "Введіть сторону першого будинку q: ";

cin >> q;

if ((p <= a) && (p <= b) && (q <= a) && (q <= b))

{

cout << "Введіть сторону другого будинку r: ";

cin >> r;

cout << "Введіть сторону другого будинку s: ";

cin >> s;

if ((r <= a) && (r <= b) && (s <= a) && (s <= b))

{

if ((q + p <= a) && (q + p <= b))//провірка будинків на перетинання між собою

{

S = a \* b;

s1 = p \* q;

s2 = r \* s;

if (S - s1 - s2 >= 0)//перевірка на вмістимість будинків на ділянці

{

cout << "Можна поставити будинки на цій ділянці " << endl;

}

else

{

cout << "Не можна поставити будинки на цій ділянці " << endl;

}

}

else

{

cout << "Будинки не можуть перетинатись один з одним " << endl;

}

}

else

{

cout << "Довжина сторона будинка2 перевищує довжина сторони ділянки " << endl;

}

}

else

{

cout << "Довжина сторона будинка1 перевищує довжина сторони ділянки " << endl;

}

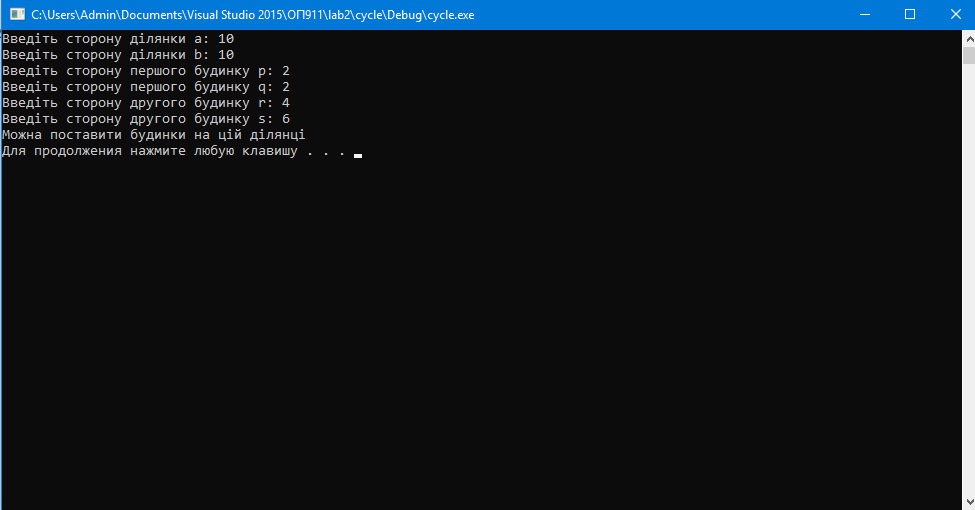
system("pause");

return;

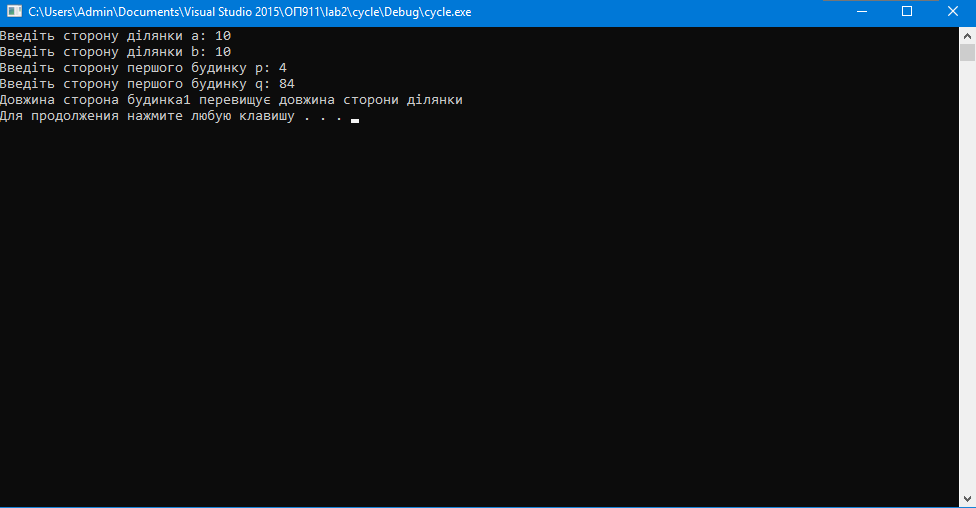
}

6. Результат роботи програми

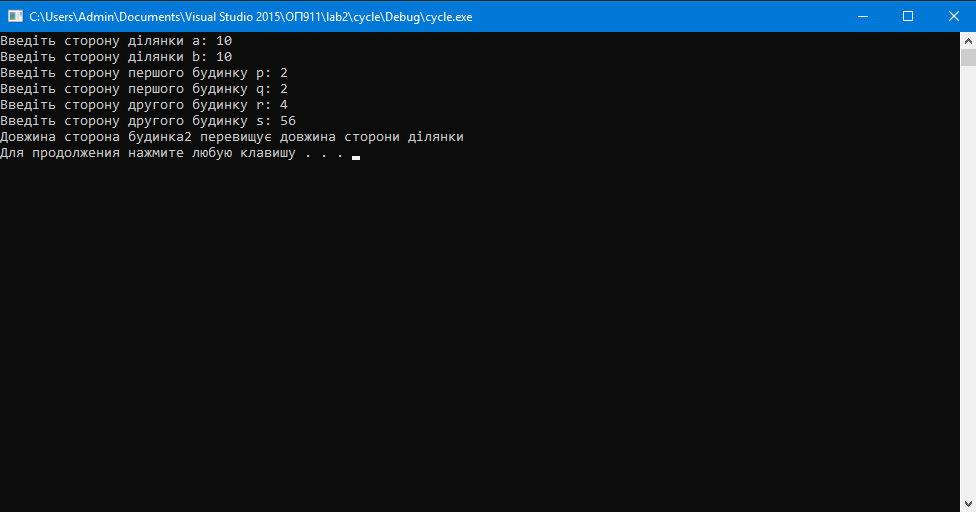
1.Правильні результати



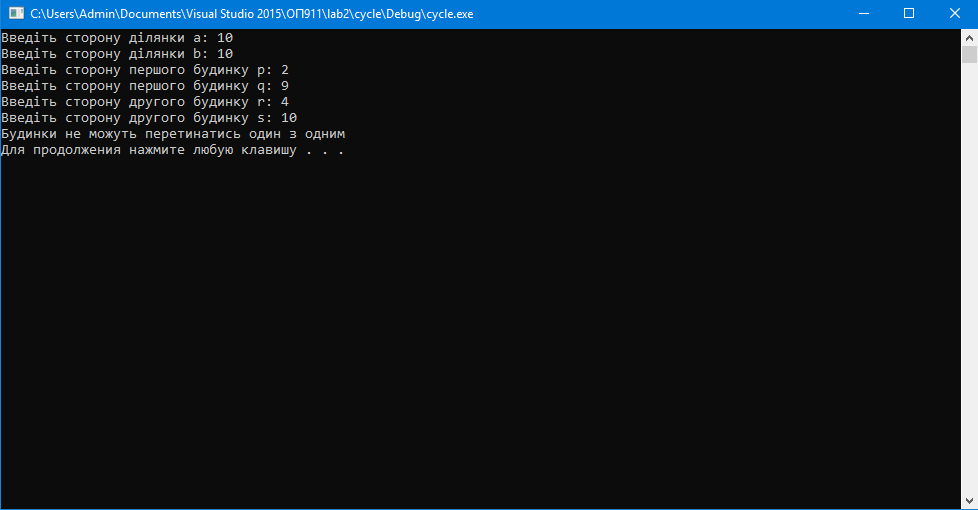
2.Велика сторона будинка1



3. Велика сторона будинка2



4.Не можуть перетинаються



7. Висновок

Розгалужений процес вміщує декілька шляхів. Вибір того чи іншого шляху залежить від виконання деяких умов. У багатьох випадках виникає потреба в зміні послідовного порядку операторів, що стає  можливим завдяки операторам управління. До них в першу чергу відносяться IF та CASE. В одній із умов я використав логічні оператори.