МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту

ЗВІТ

**РОЗРАХУНКОВА РОБОТА**

З дисципліни:

**Дискретна математика**

Виконав

Студент групи **КН-113**

**Макогін Назарій**

Викладач:

**Мельникова Н.І.**

**Львів 2019р**

**Варіант 11:**

**Завдання № 1**

Виконати наступні операції над графами:

1) знайти доповнення до першого графу,

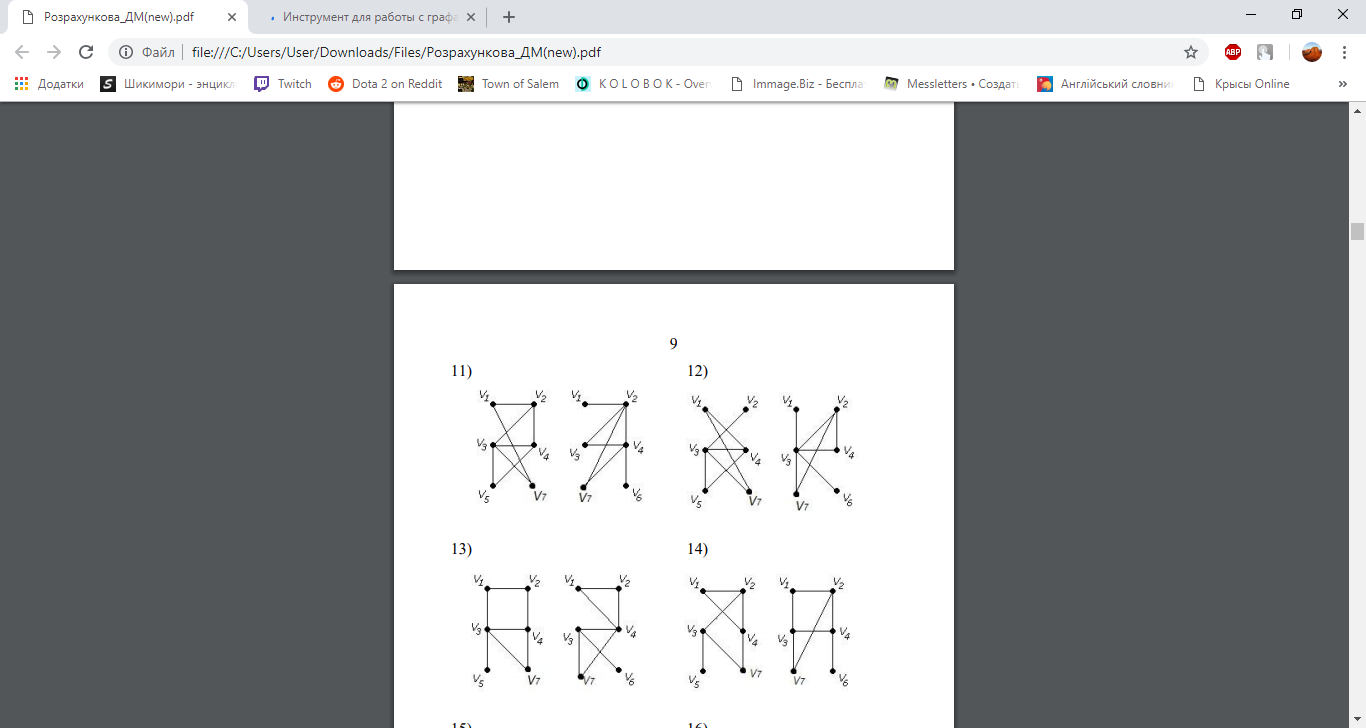
2) об’єднання графів,

3) кільцеву сумму G1 та G2 (G1+G2),

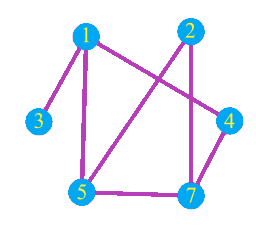
4) розмножити вершину у другому графі,

5) виділити підграф А - що скадається з 3-х вершин в G1 та виконати G1\A

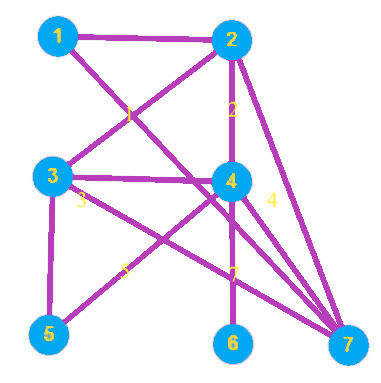
6) добуток графів.



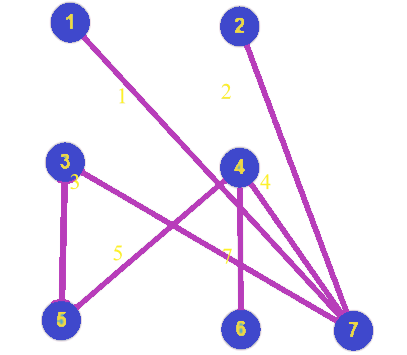
1.1) Доповнення до першого графа:



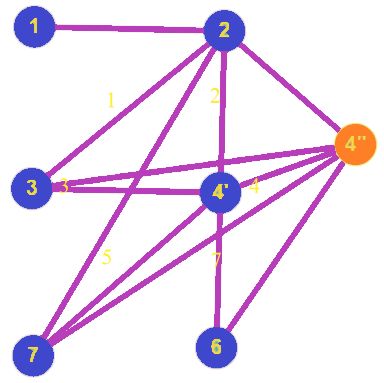
1.2) Об’єднання графів:



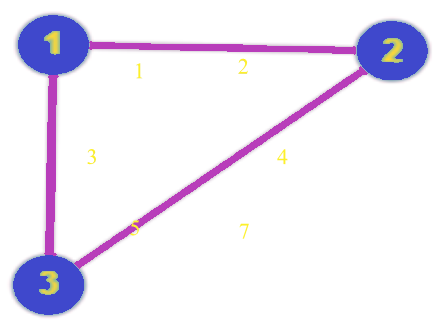
1.3) Кільцева сума:



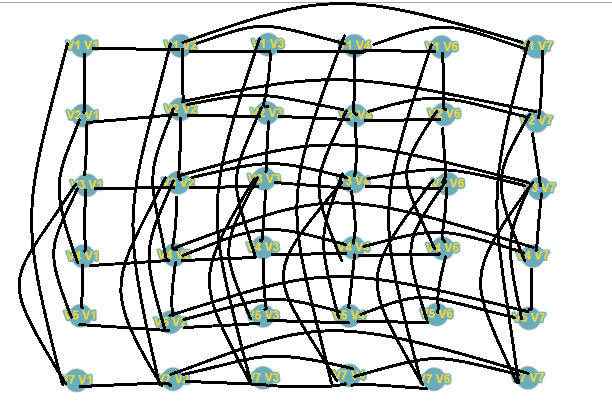
1.4) Розщеплення вершини у другому графі:



1.5) Стягнення підграфу А(V4 V5 V7)до вершини V3 у G1:

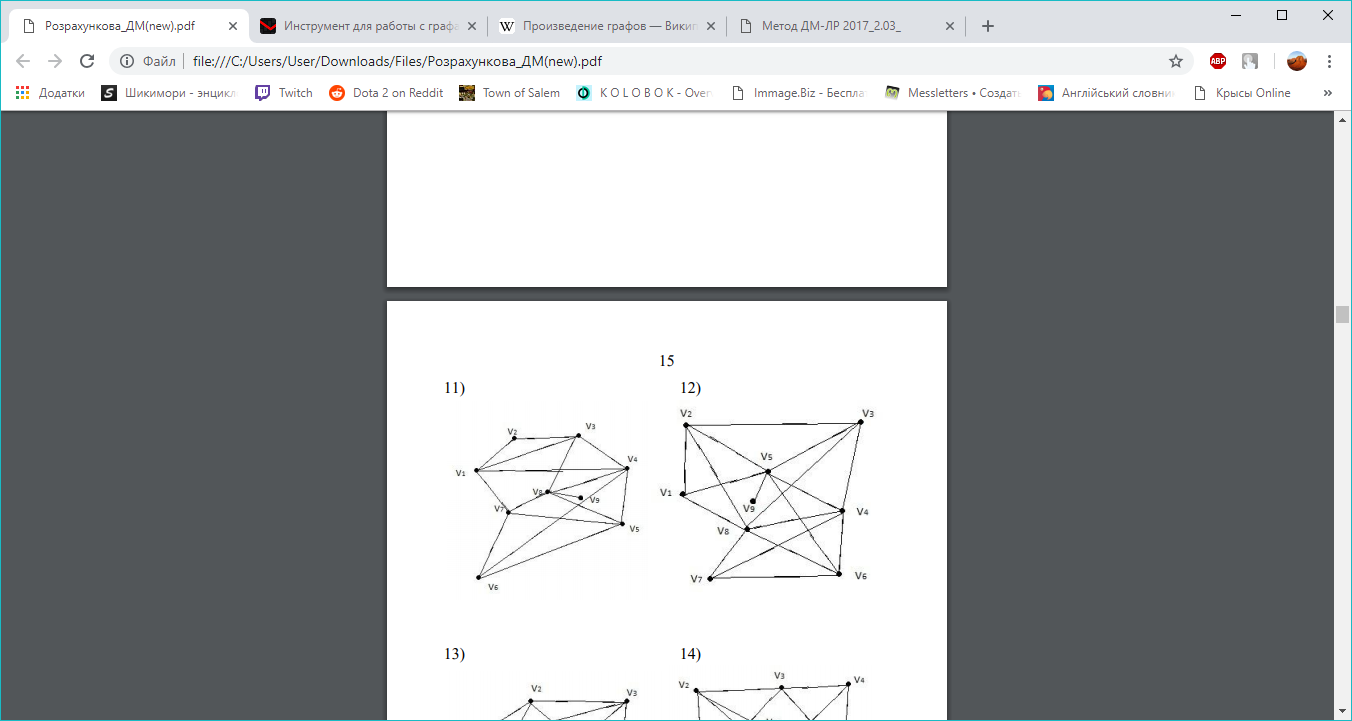


1.6) Добуток графів:



**Завдання № 2**

Скласти таблицю суміжності для орграфа.



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 |
| V1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| V2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| V3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| V4 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| V5 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| V6 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| V7 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| V8 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| V9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |

**Завдання №3:**

Для графа з другого завдання знайти діаметр.

3.Діаметр графа: 3 (v2->v1->v7->v6)

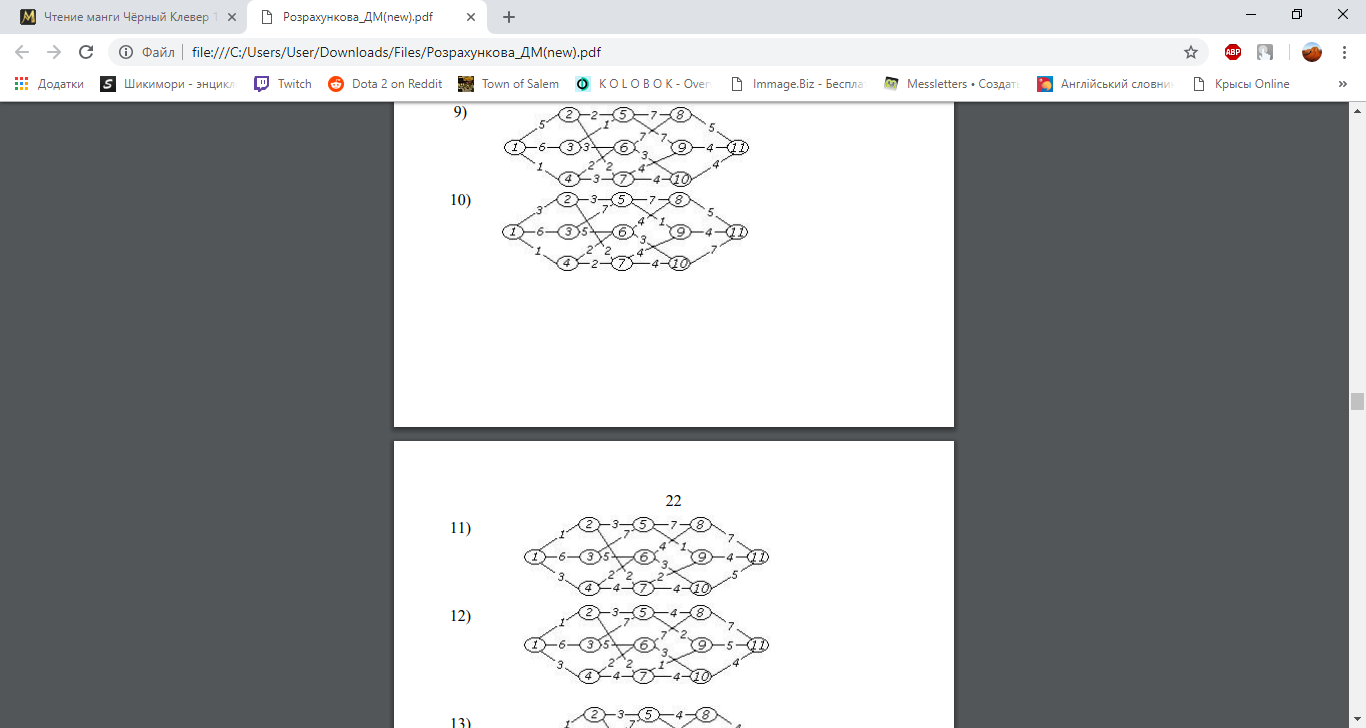
**Завдання №4:**

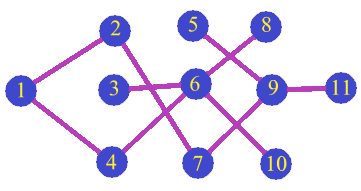
Для графа з другого завдання виконати обхід дерева вглиб (варіант закінчується на непарне число) або вшир (закінчується на парне число).

|  |  |  |
| --- | --- | --- |
| Верш | DFS | Стек |
| V1 | 1 | V2V3V4V7 |
| V2 | 2 | V3V4V7 |
| V3 | 3 | V8V4V8 |
| V4 | 4 | V1V2V3V4 |
| V5 | 5 | V1V2V3V4V5 |
| V6 | 6 | V1V2V3V4V5V6 |
| V7 | 7 | V1V2V3V4V5V6V7 |
| V8 | 8 | V1V2V3V4V5V6V7V8 |
| V9 | 9 | V1V2V3V4V5V6V7V8V9 |
|  |  |  |

**Завдання № 5**

Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.





**За прима: За краскала:**

1-2 1 1-2 1

2-7 2 9-5 1

7-9 2 2-7 2

9-5 1 7-9 2

1-4 3 4-6 2

4-6 2 6-10 2

6-10 2 1-4 3

6-8 4 6-8 4

9-11 4 9-11 4

3-6 5 3-6 5

***Програма до краскала:***

#include <iostream>

#include <stdio.h>

using namespace std;

struct edge {

int leng;

int p1;

int p2;

bool in = false;

};

struct mas {

int arr[11] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

int c = 0;

};

int in(edge \*reb, int n) {

setlocale(LC\_ALL, "Ukrainian");

for (int i = 0; i < n; i++)

{

cout << "Введiть довжину " << i + 1 << " ребра: ";

cin >> reb[i].leng;

cout << "Введiть першу сумiжну вершину з " << i + 1 << " ребром: ";

cin >> reb[i].p1;

cout << "Введiть другу сумiжну вершину з " << i + 1 << " ребром: ";

cin >> reb[i].p2;

cout << endl;

}

return 0;

}

int main() {

setlocale(LC\_ALL, "Ukrainian");

int n = 0, x = 100, y = 100;

cout << "Введiть кiлькiсть ребер у графi: ";

cin >> n;

int z;

cout << "Введiть кiлькiсть вершин у графi: ";

cin >> z;

cout << endl;

edge \*reb = new edge[n];

mas inn[8];

in(reb, n);

for (int i = 0; i < n - 1; i++)

{

for (int j = 0; j < n - 1; j++)

{

if (reb[j].leng > reb[j + 1].leng) { swap(reb[j].leng, reb[j + 1].leng); swap(reb[j].p1, reb[j + 1].p1); swap(reb[j].p2, reb[j + 1].p2); }

}

}

int c = -1;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < 8; j++)

{

for (int k = 0; k < z; k++)

{

if (reb[i].p1 == inn[j].arr[k]) { x = j; goto point0;; }

}

}

point0:;

for (int j = 0; j < 8; j++)

{

for (int k = 0; k <z; k++)

{

if (reb[i].p2 == inn[j].arr[k]) { y = j; goto point1; }

}

}

point1:;

if (x != y && x == 100) { inn[y].arr[inn[y].c] = reb[i].p1; inn[y].c++; }

if (x != y && y == 100) { inn[x].arr[inn[x].c] = reb[i].p2; inn[x].c++; }

if (x != y && x != 100 && y != 100) {

if (x < y) {

for (int l = 0; l < inn[y].c; l++)

{

inn[x].arr[inn[x].c+l] = inn[y].arr[l];

inn[y].arr[l] = 0;

}

inn[x].c += inn[y].c;

}

if (y < x) {

for (int l = 0; l < inn[y].c; l++)

{

inn[y].arr[inn[y].c+l] = inn[x].arr[l];

inn[x].arr[l] = 0;

}

inn[y].c += inn[x].c;

}

}

if (x == 100 && y == 100) { c++; inn[c].arr[inn[c].c] = reb[i].p1; inn[c].arr[inn[c].c + 1] = reb[i].p2; inn[c].c += 2; }

reb[i].in = true;

if (x == y && x != 100) { reb[i].in = false; }

x = 100; y = 100;

}

cout << "ребра остового дерева: " << endl;

int v = 0, s = 0;;

for (int i = 0; i < n; i++)

{

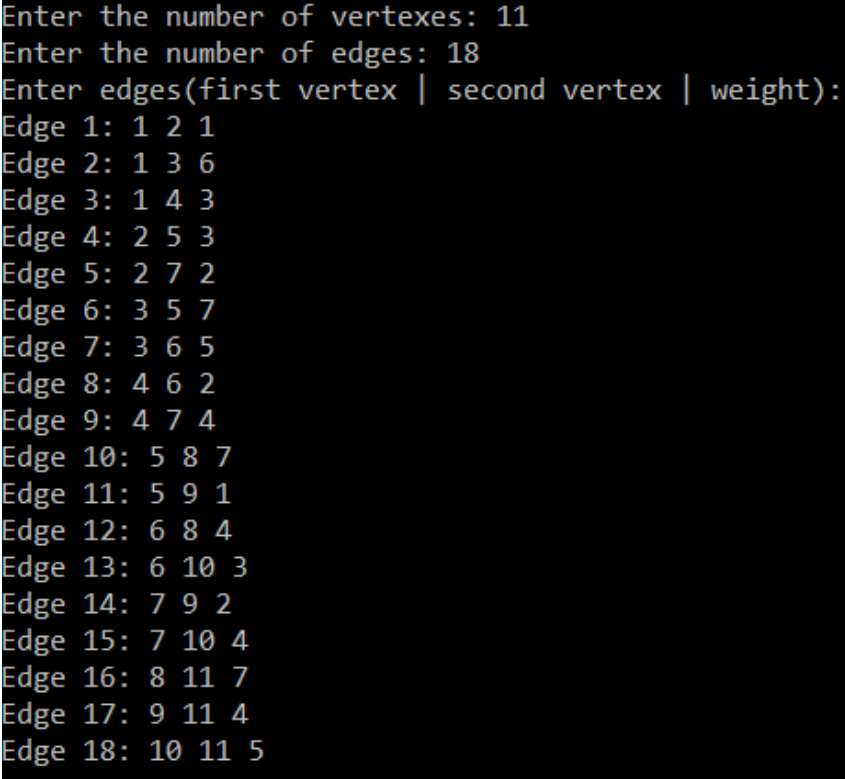
if (reb[i].in == true) { cout << "Ребро" << ", що сполучає вершини " << reb[i].p1 << " " << reb[i].p2 << endl; v++; s += reb[i].leng; }

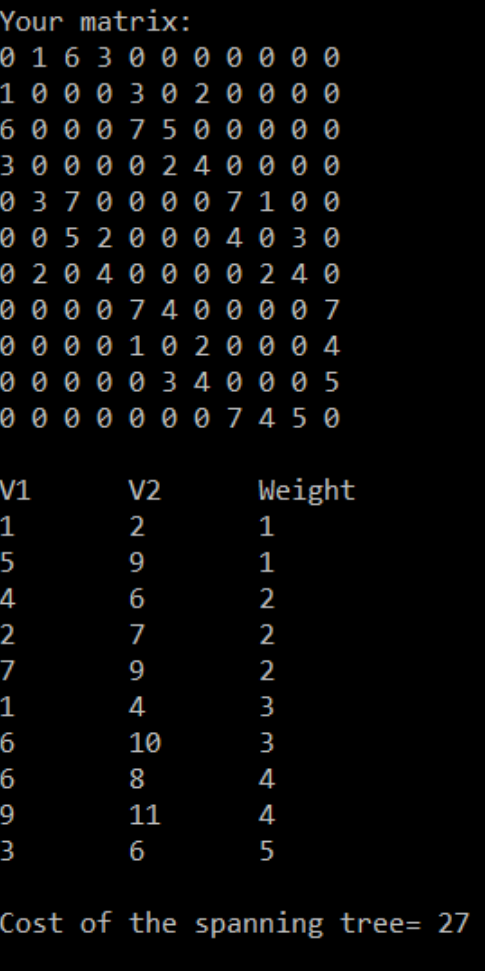
if (v == z-1) { break; }

}

cout << "Остове дерево мiнiмальної ваги для даного графа: " << s;

return 0;}





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

***Програма до прима:***

#include <iostream>

#include <iomanip>

using namespace std;

void output(int size, int\*\* adjacencyarr) {

cout << " ";

for (int i = 0; i < size; i++) { cout << setw(3) << i + 1; }

for (int i = 0; i < size; i++) {

cout << endl << setw(3) << i + 1;

for (int j = 0; j < i; j++) {

cout << setw(3) << adjacencyarr[j][i];

}

cout << " -";

for (int j = i + 1; j < size; j++) {

cout << setw(3) << adjacencyarr[i][j];

}

}

}

void main()

{

int size;

cout << "Enter amount of vertices ";

cin >> size;

int\*\* adjacencyarr = new int\* [size];

bool\* vertex = new bool[size];

cout << "Enter adjacency matrix \n";

cout << " ";

for (int i = 0; i < size; i++) { cout << setw(3) << i + 1; adjacencyarr[i] = new int[size]; vertex[i] = 0; }

vertex[0] = 1;

cout << endl;

for (int i = 0; i < size; i++) {

cout << setw(2) << i + 1;

for (int j = 0; j < i; j++) { cout << setw(3) << adjacencyarr[j][i]; }

cout << " -";

for (int j = i + 1; j < size; j++) { cin >> adjacencyarr[i][j]; }

}

cout << endl;

int min, minI, minJ;

for (int n = 0; n < size - 1; n++) {

min = 100, minI = 0, minJ = 0;

for (int i = 0; i < size; i++) {

for (int j = i + 1; j < size; j++) {

if (adjacencyarr[i][j] < min && adjacencyarr[i][j] != 0 && vertex[i]!=vertex[j]) {

min = adjacencyarr[i][j];

minI = i, minJ = j;

}

}

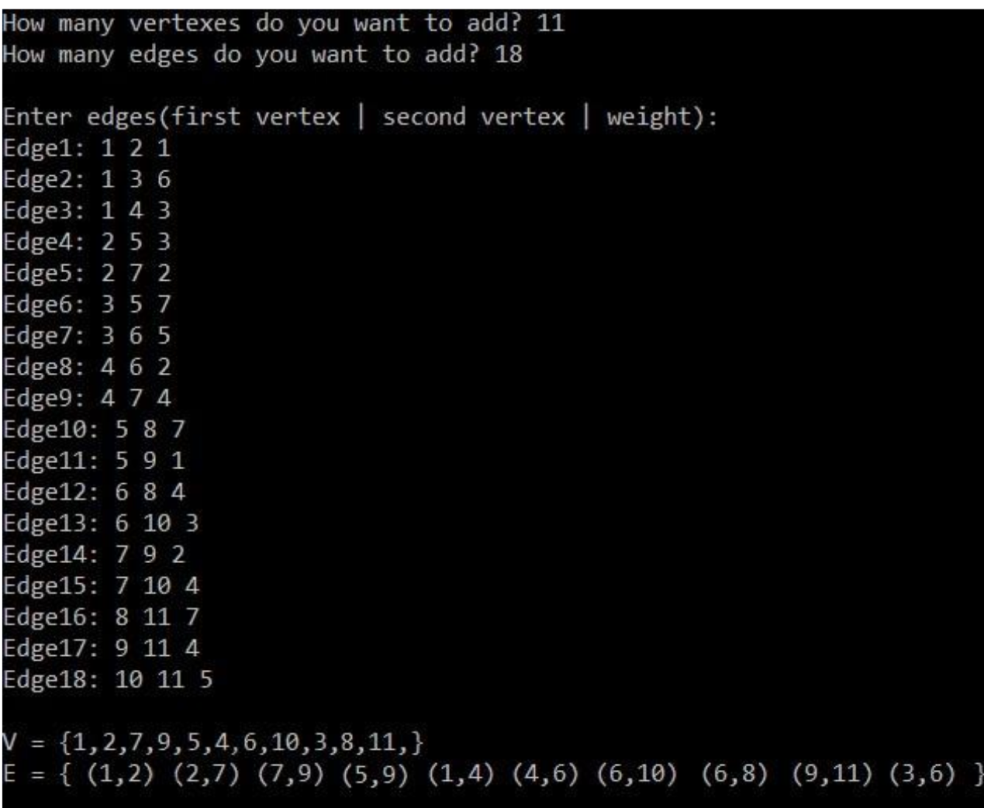
}

vertex[minI] = 1; vertex[minJ] = 1;

cout << n + 1 << ")conected " << minI + 1 << "-" << minJ + 1 << endl;

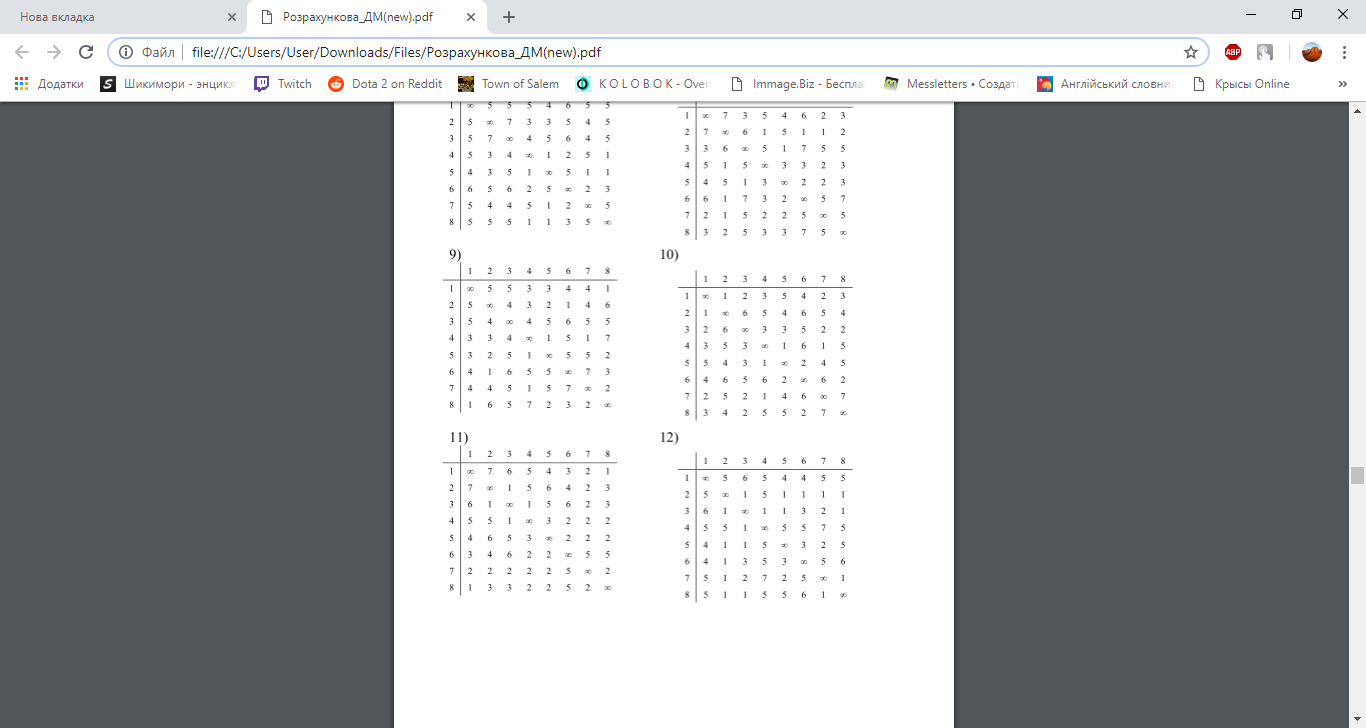
}

}



**Завдання № 6**

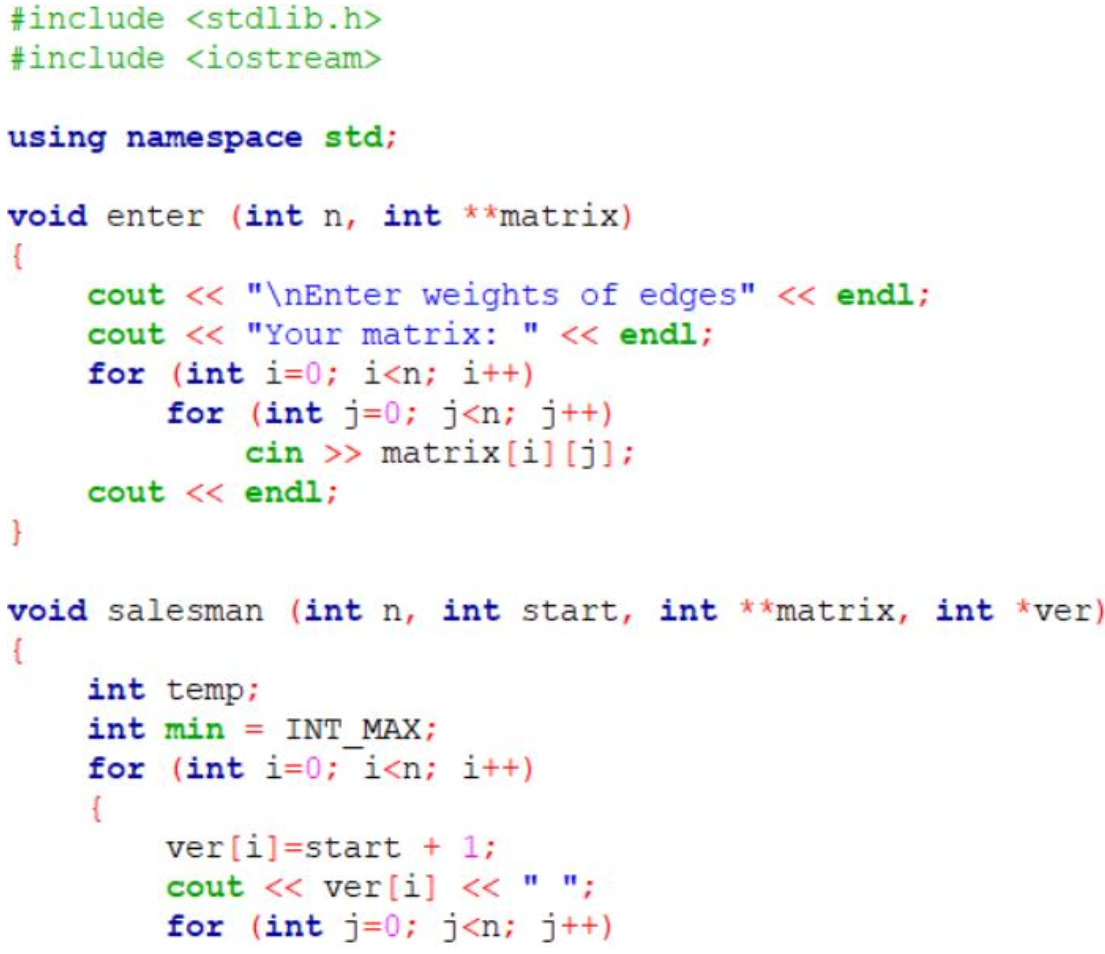
Розв’язати задачу комiвояжера для повного 8-ми вершинного графа методом «iди у найближчий», матриця вагiв якого має вигляд:

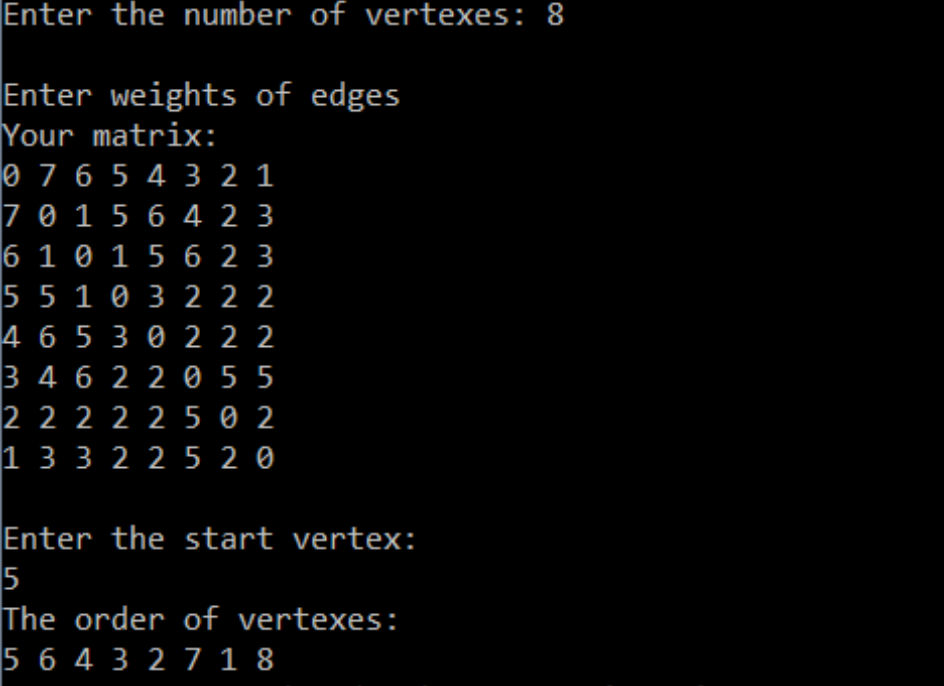
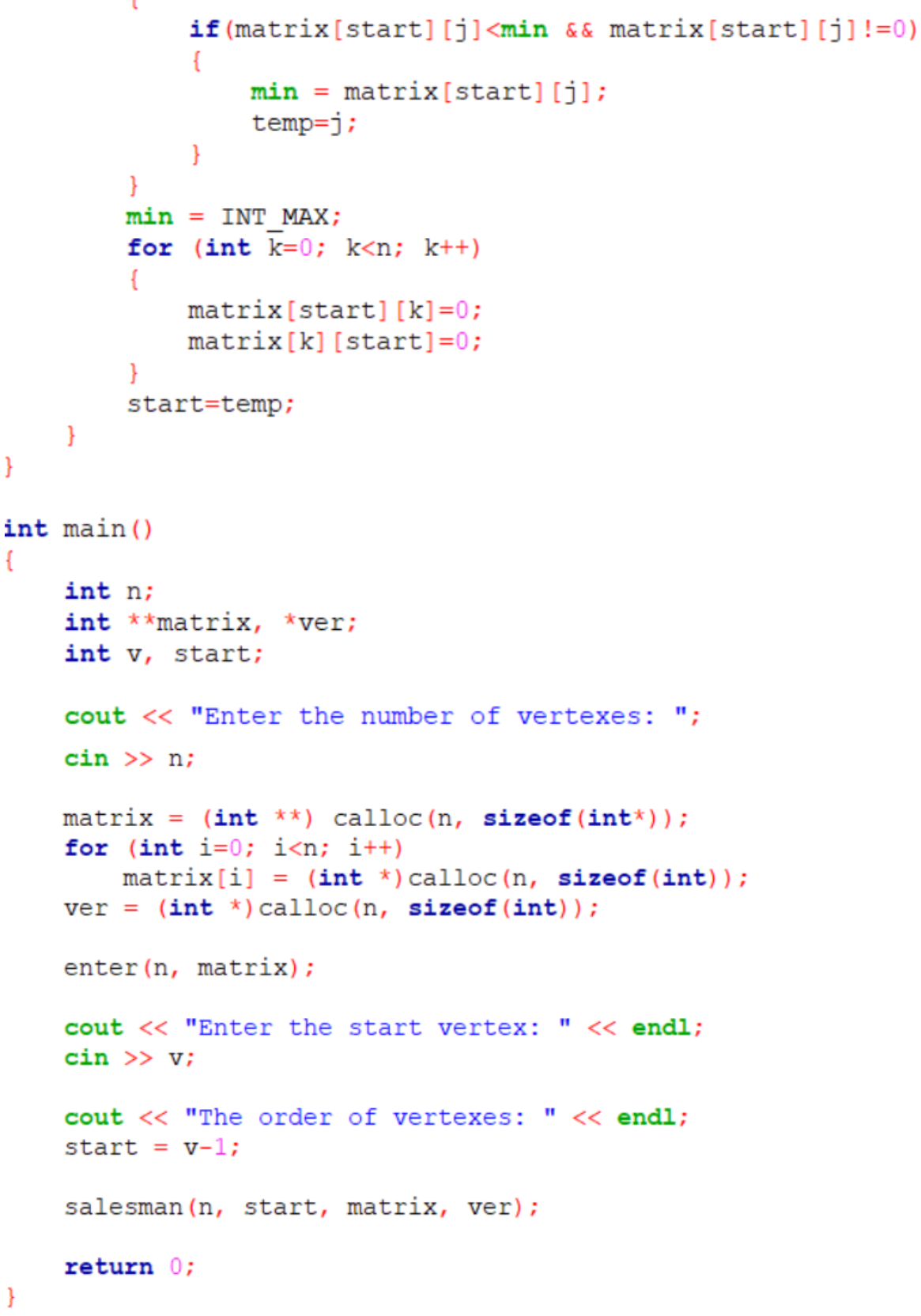


S – (1,8),(8,4),(4,3),(3,2),(2,7),(7,5),(5,6),(6,1)

Довжина маршруту 1+2+1+1+2+2+2+3 = 14

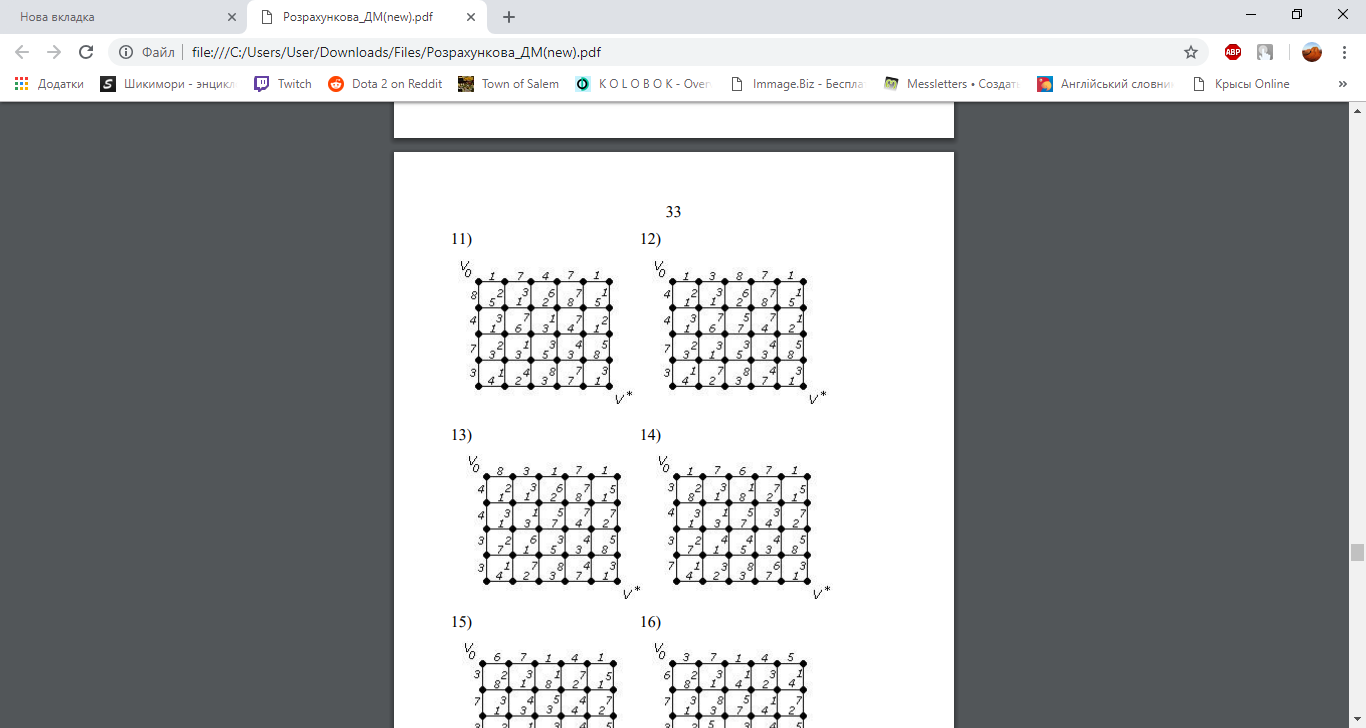
***Програма:***

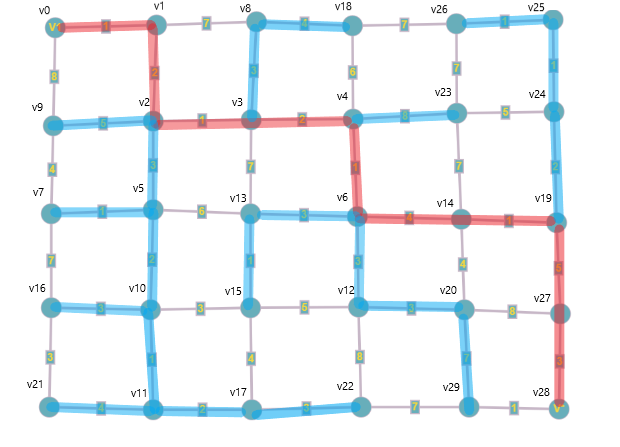




**Завдання № 7**

За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі між парою вершин V0 і V \* .





L(V0) = 0 L(V1) = 1 L(V2) = 3 L(V3) = 4 L(V4) = 6 L(V5) = 6

L(V6) = 7 L(V7) = 7 L(V8) = 7 L(V9) = 8 L(V10) = 8 L(V11) = 9

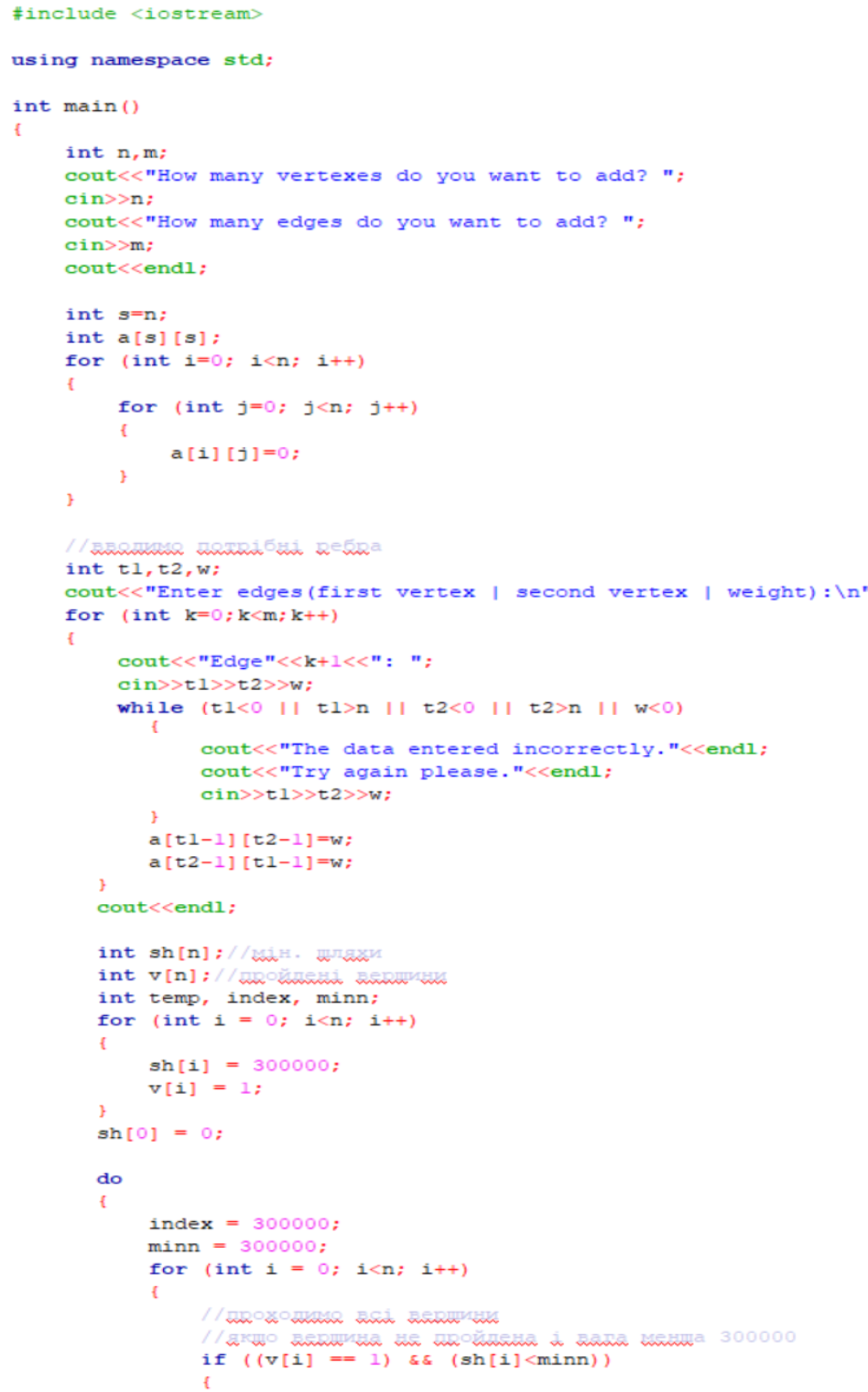
L(V12) = 10 L(V13) = 10 L(V14) = 11 L(V15) = 11 L(V16) = 11 L(V17) = 11

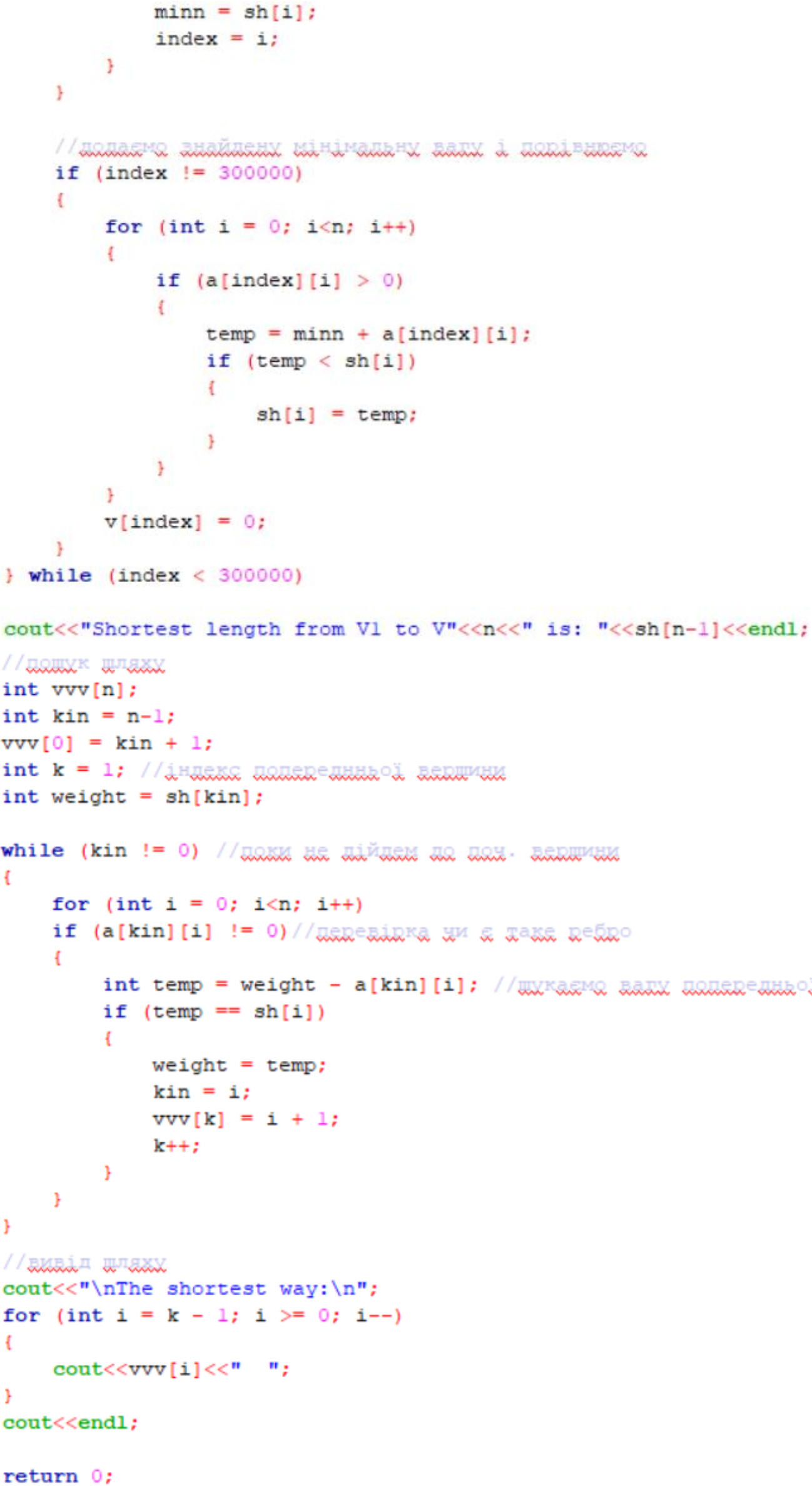
L(V18) = 11 L(V19) = 12 L(V20) = 13 L(V21) = 13 L(V22) = 14 L(V23) = 14

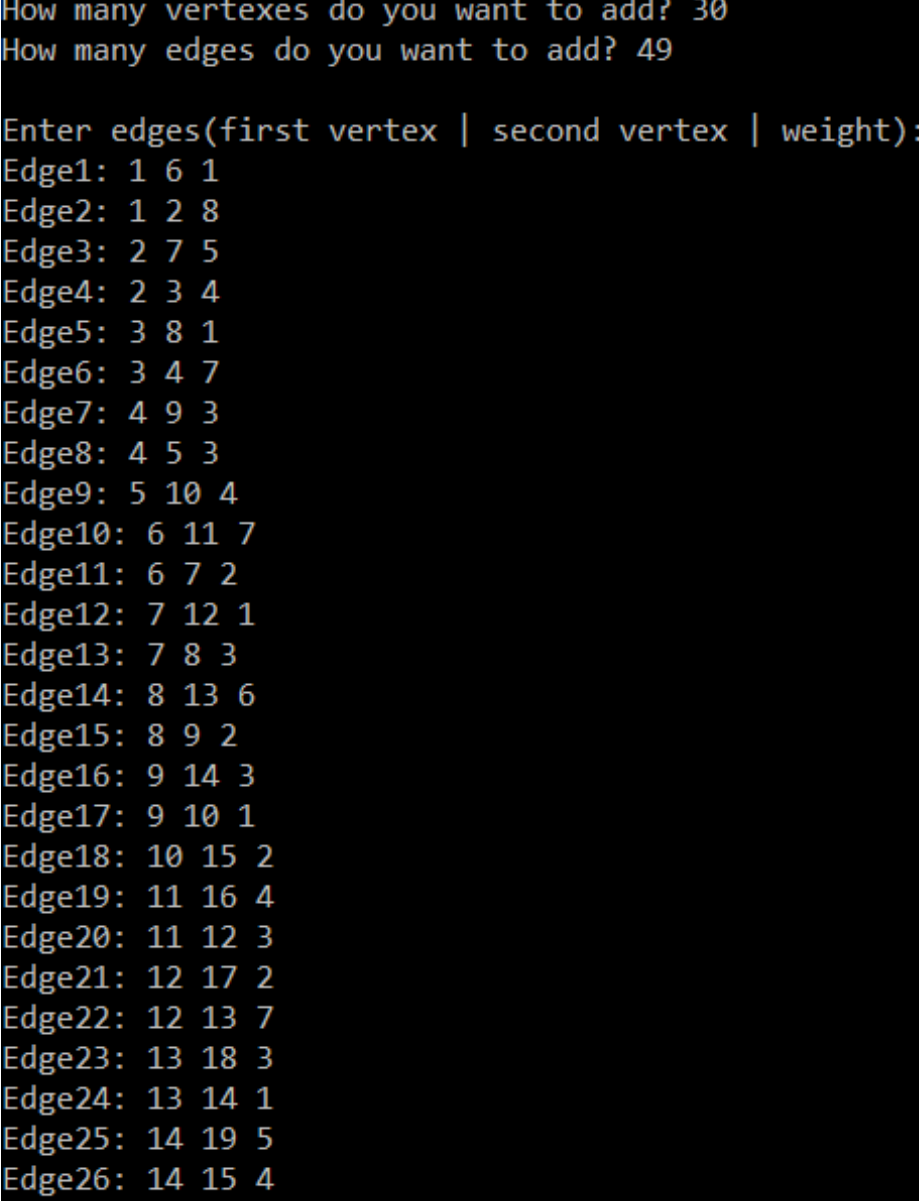
L(V24) = 14 L(V25) = 15 L(V26) = 16 L(V27) = 17 L(V28) = 20 L(V29) = 20

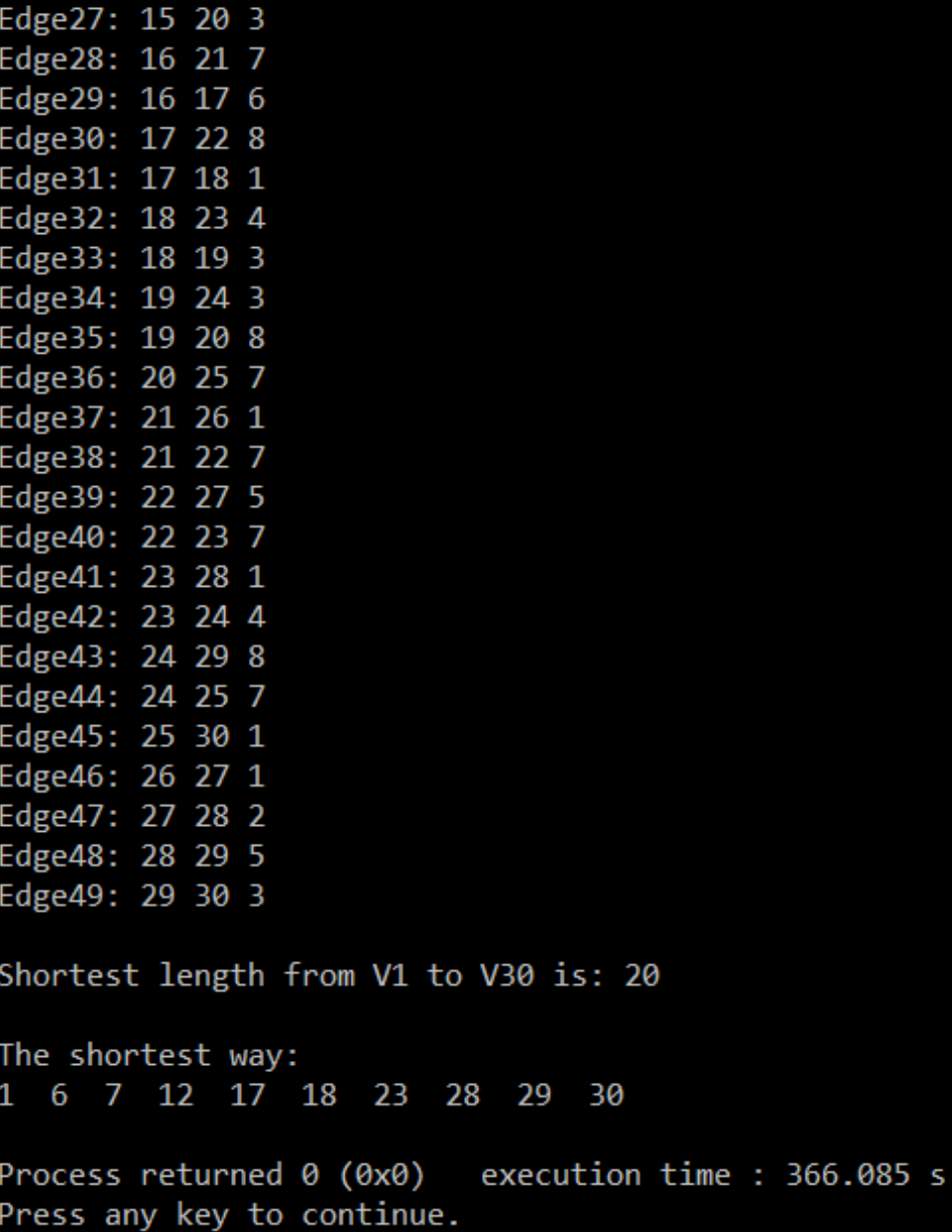
Найкоротша відстань між заданим вершинами S { V0 -> V1 -> V2 -> V3 - >V4 -> V6 -> V14 -> V17 -> V27 -> V28} = 20 позначена червоним кольором.

***Програма:***



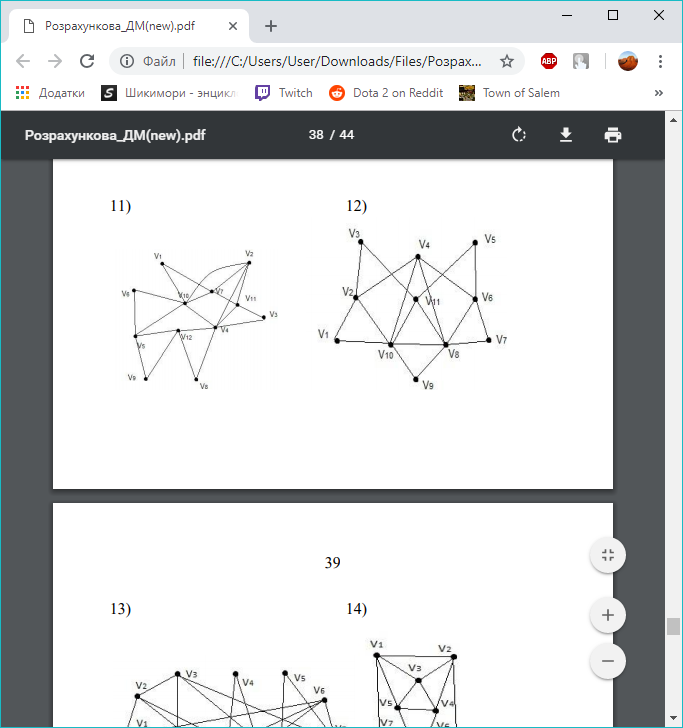






**Завдання № 8**

Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.



А) За методом Флері

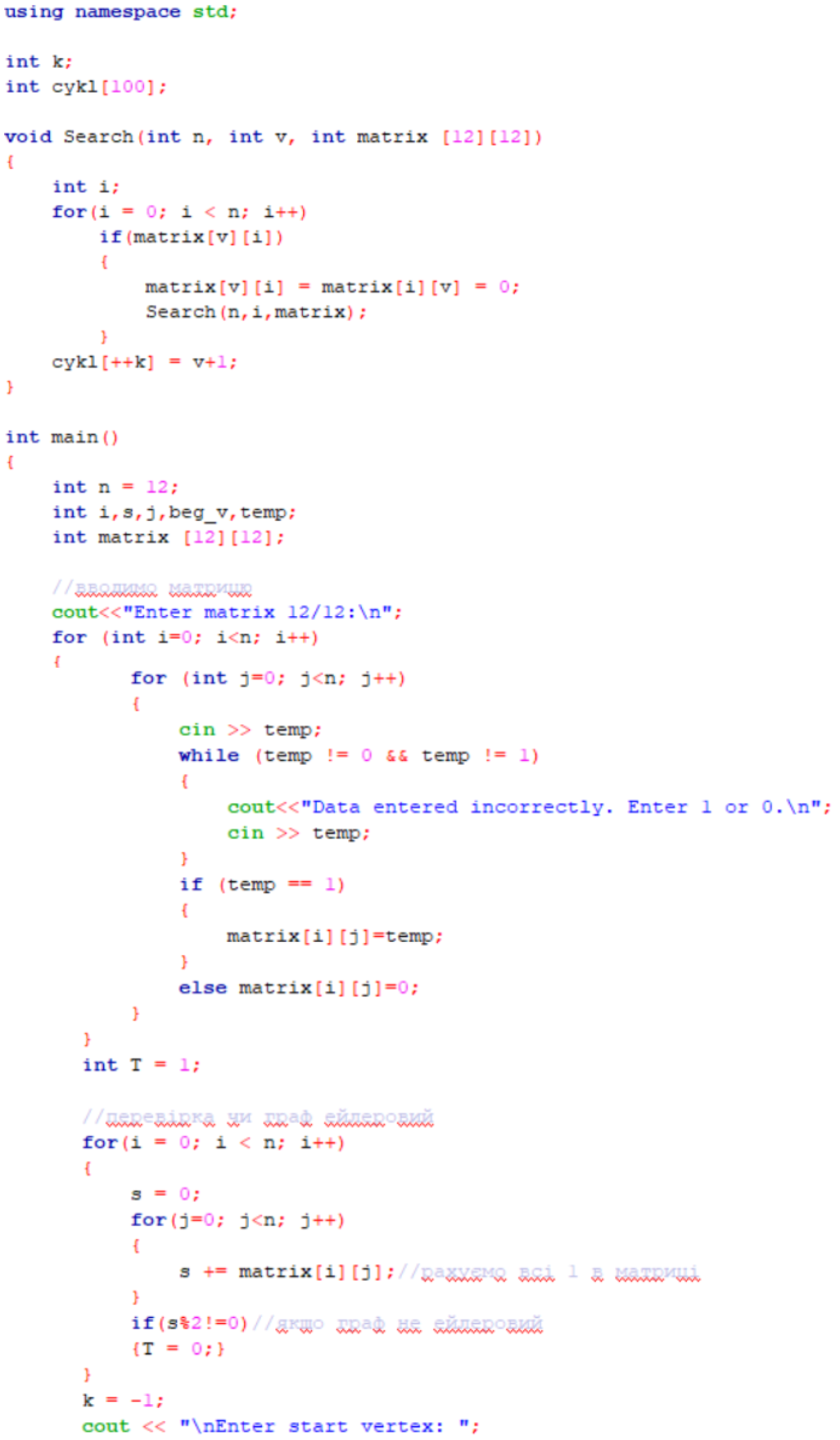
(1,7),(7,2),(2,4),(4,11),(11,2),(2,10),(10,6),(6,5),(5,12),(12,9),(9,5),(5,10),(7,11),(11,3)

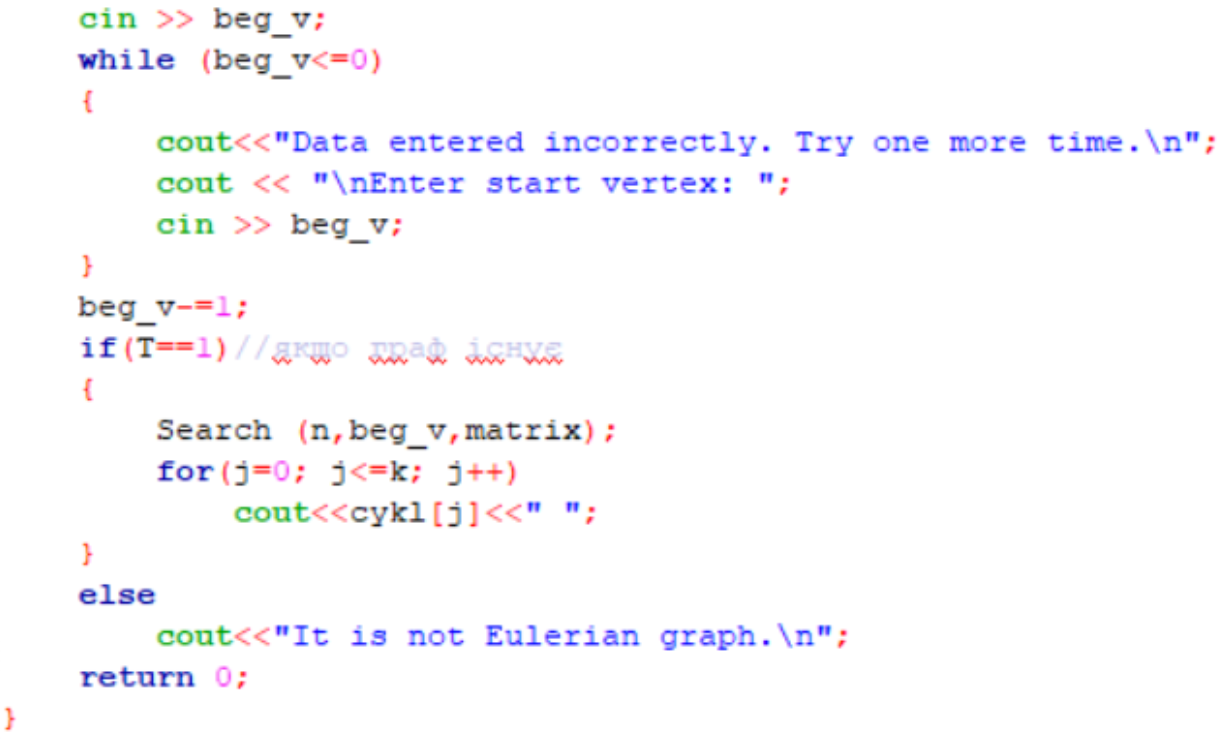
(3,4),(4,12),(12,8),(8,4),(4,10),(10,1)

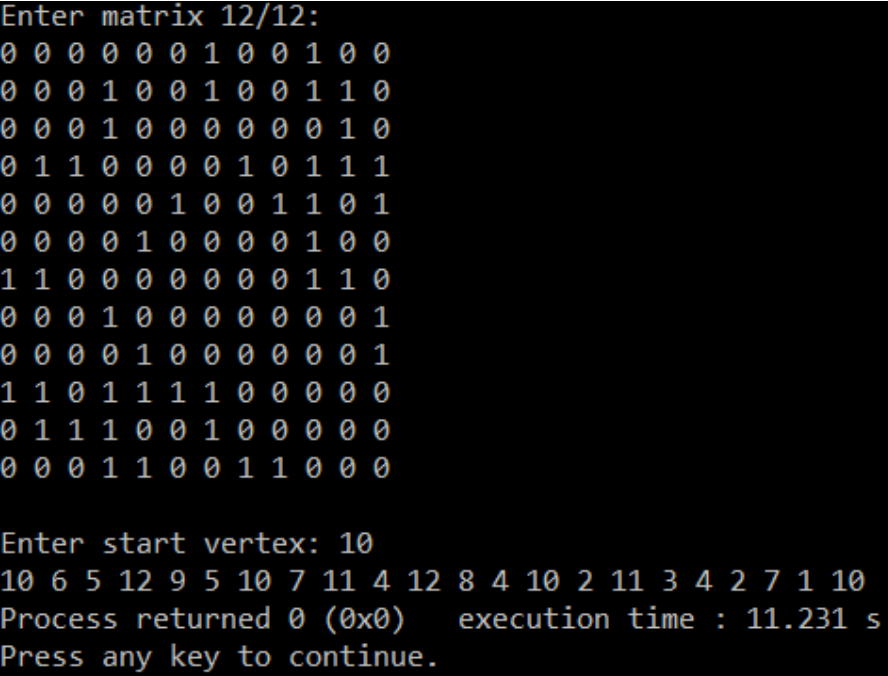
Б) За методом елементарних циклів

(1->7->2->10->1),(1->7->11->2->10->1),(1->7->11->3->4->2->10->1),(1->7->11->4->2->10->1),(1->7->2->4->10->1),(1->7->11->2->4->10->1),(1->7->2->11->3->4->10->1),(1->7->11->3->4->10->1),(1->7->2->11->4->10->1),(1->7->11->4->10->1),(1->7->2->4->12->9->5->10->1),(1->7->11->2->4->12->9->5->10->1),(1->7->2->11->3->4->12->9->5->10->1),(1->7->11->3->4->12->9->5->10->1),(1->7->2->11->4->12->9->5->10->1),(1->7->11->4->12->9->5->10->1),(1->7->2->4->8->12->9->5->10->1),(1->7->11->2->4->8->12->9->5->10->1),(1->7->2->11->3->4->8->12->9->5->10->1),(1->7->11->3->4->8->12->9->5->10->1),(1->7->2->11->4->8->12->9->5->10->1),(1->7->11->4->8->12->9->5->10->1),(1->7->2->4->12->5->10->1),(1->7->11->2->4->12->5->10->1),(1->7->2->11->3->4->12->5->10->1),(1->7->11->3->4->12->5->10->1),(1->7->2->11->4->12->5->10->1),(1->7->11->4->12->5->10->1),(1->7->2->4->8->12->5->10->1),(1->7->11->2->4->8->12->5->10->1),(1->7->2->11->3->4->8->12->5->10->1),(1->7->11->3->4->8->12->5->10->1),(1->7->2->11->4->8->12->5->10->1),(1->7->11->4->8->12->5->10->1),(1->7->2->4->12->9->5->6->10->1),(1->7->11->2->4->12->9->5->6->10->1),(1->7->2->11->3->4->12->9->5->6->10->1),(1->7->11->3->4->12->9->5->6->10->1),(1->7->2->11->4->12->9->5->6->10->1),(1->7->11->4->12->9->5->6->10->1),(1->7->2->4->8->12->9->5->6->10->1),(1->7->11->2->4->8->12->9->5->6->10->1),(1->7->2->11->3->4->8->12->9->5->6->10->1),(1->7->11->3->4->8->12->9->5->6->10->1),(1->7->2->11->4->8->12->9->5->6->10->1),(1->7->11->4->8->12->9->5->6->10->1),(1->7->2->4->12->5->6->10->1),(1->7->11->2->4->12->5->6->10->1),(1->7->2->11->3->4->12->5->6->10->1),(1->7->11->3->4->12->5->6->10->1),(1->7->2->11->4->12->5->6->10->1),(1->7->11->4->12->5->6->10->1),(1->7->2->4->8->12->5->6->10->1),(1->7->11->2->4->8->12->5->6->10->1),(1->7->2->11->3->4->8->12->5->6->10->1),(1->7->11->3->4->8->12->5->6->10->1),(1->7->2->11->4->8->12->5->6->10->1),(1->7->11->4->8->12->5->6->10->1),(1->7->10->1),(2->10->7->1),(2->10->4->3->11->7->1),(2->10->6->5->12->8->4->3->11->7->1),(2->10->5->12->8->4->3->11->7->1),(2->10->6->5->9->12->8->4->3->11->7->1),(2->10->5->9->12->8->4->3->11->7->1),(2->10->6->5->12->4->3->11->7->1),(2->10->5->12->4->3->11->7->1),(2->10->6->5->9->12->4->3->11->7->1),(2->10->5->9->12->4->3->11->7->1),(2->10->4->11->7->1),(2->10->6->5->12->8->4->11->7->1),(2->10->5->12->8->4->11->7->1),(2->10->6->5->9->12->8->4->11->7->1),(2->10->5->9->12->8->4->11->7->1),(2->10->6->5->12->4->11->7->1),(2->10->5->12->4->11->7->1),(2->10->6->5->9->12->4->11->7->1),(2->10->5->9->12->4->11->7->1),(2->10->4->3->11->1),(2->10->6->5->12->8->4->3->11->1),(2->10->5->12->8->4->3->11->1),(2->10->6->5->9->12->8->4->3->11->1),(2->10->5->9->12->8->4->3->11->1),(2->10->6->5->12->4->3->11->1),(2->10->5->12->4->3->11->1),(2->10->6->5->9->12->4->3->11->1),(2->10->5->9->12->4->3->11->1),(2->10->4->11->1),(2->10->6->5->12->8->4->11->1),(2->10->5->12->8->4->11->1),(2->10->6->5->9->12->8->4->11->1),(2->10->5->9->12->8->4->11->1),(2->10->6->5->12->4->11->1),(2->10->5->12->4->11->1),(2->10->6->5->9->12->4->11->1),(2->10->5->9->12->4->11->1),(2->10->7->11->1),(2->10->7->11->3->4->1),(2->10->7->11->4->1),(2->10->4->1),(2->10->6->5->12->8->4->1),(2->10->5->12->8->4->1),(2->10->6->5->9->12->8->4->1),(2->10->5->9->12->8->4->1),(2->10->6->5->12->4->1),(2->10->5->12->4->1),(2->10->6->5->9->12->4->1),(2->10->5->9->12->4->1),(2->4->10->7->1),(2->4->10->7->11->1),(2->11->3->4->10->7->1),(3->4->10->7->11->1),(2->11->4->10->7->1),(4->10->7->11->1),(4->10->6->5->12->8->1),(4->10->5->12->8->1),(4->10->6->5->9->12->8->1),(4->10->5->9->12->8->1),(4->10->6->5->12->1),(4->10->5->12->1),(4->10->6->5->9->12->1),(4->10->5->9->12->1),(2->4->12->9->5->10->7->1),(2->4->12->9->5->10->7->11->1),(2->11->3->4->12->9->5->10->7->1),(3->4->12->9->5->10->7->11->1),(2->11->4->12->9->5->10->7->1),(4->12->9->5->10->7->11->1),(2->4->8->12->9->5->10->7->1),(2->4->8->12->9->5->10->7->11->1),(2->11->3->4->8->12->9->5->10->7->1),(3->4->8->12->9->5->10->7->11->1),(2->11->4->8->12->9->5->10->7->1),(4->8->12->9->5->10->7->11->1),(2->4->12->5->10->7->1),(2->4->12->5->10->7->11->1),(2->11->3->4->12->5->10->7->1),(3->4->12->5->10->7->11->1),(2->11->4->12->5->10->7->1),(4->12->5->10->7->11->1),(2->4->8->12->5->10->7->1),(2->4->8->12->5->10->7->11->1),(2->11->3->4->8->12->5->10->7->1),(3->4->8->12->5->10->7->11->1),(2->11->4->8->12->5->10->7->1),(4->8->12->5->10->7->11->1),(5->10->6->1),(2->4->12->9->5->6->10->7->1),(2->4->12->9->5->6->10->7->11->1),(2->11->3->4->12->9->5->6->10->7->1),(3->4->12->9->5->6->10->7->11->1),(2->11->4->12->9->5->6->10->7->1),(4->12->9->5->6->10->7->11->1),(2->4->8->12->9->5->6->10->7->1),(2->4->8->12->9->5->6->10->7->11->1),(2->11->3->4->8->12->9->5->6->10->7->1),(3->4->8->12->9->5->6->10->7->11->1),(2->11->4->8->12->9->5->6->10->7->1),(4->8->12->9->5->6->10->7->11->1),(2->4->12->5->6->10->7->1),(2->4->12->5->6->10->7->11->1),(2->11->3->4->12->5->6->10->7->1),(3->4->12->5->6->10->7->11->1),(2->11->4->12->5->6->10->7->1),(4->12->5->6->10->7->11->1),(2->4->8->12->5->6->10->7->1),(2->4->8->12->5->6->10->7->11->1),(2->11->3->4->8->12->5->6->10->7->1),(3->4->8->12->5->6->10->7->11->1),(2->11->4->8->12->5->6->10->7->1),(4->8->12->5->6->10->7->11->1),(2->7->11->1),(2->7->11->3->4->1),(2->7->11->4->1),(2->4->3->11->1),(2->4->11->1),(3->11->4->1),(4->12->8->1),(5->12->9->1),( 1->7->2->10->1)

***Програма:***

****

****

****

**Завдання №9**

Спростити формули (привести їх до скороченої ДНФ).

x̅ y ∨ y̅ ≥ x;

x̅ ∨ x = T;

y ∨ y̅ = T;

T ∨ z̅ = z̅;