Міністерство освіти і науки України

Вінницький національний технічний університет

Інститут інформаційних технологій та комп’ютерної інженерії

Кафедра комп’ютерних наук

Лабораторна робота № 6

З дисципліни: “Дискретна математика”

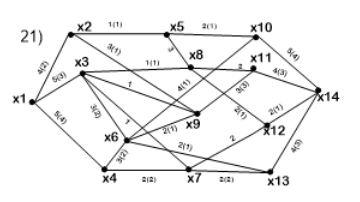
Тема: “ Побудова стовбура графа ”

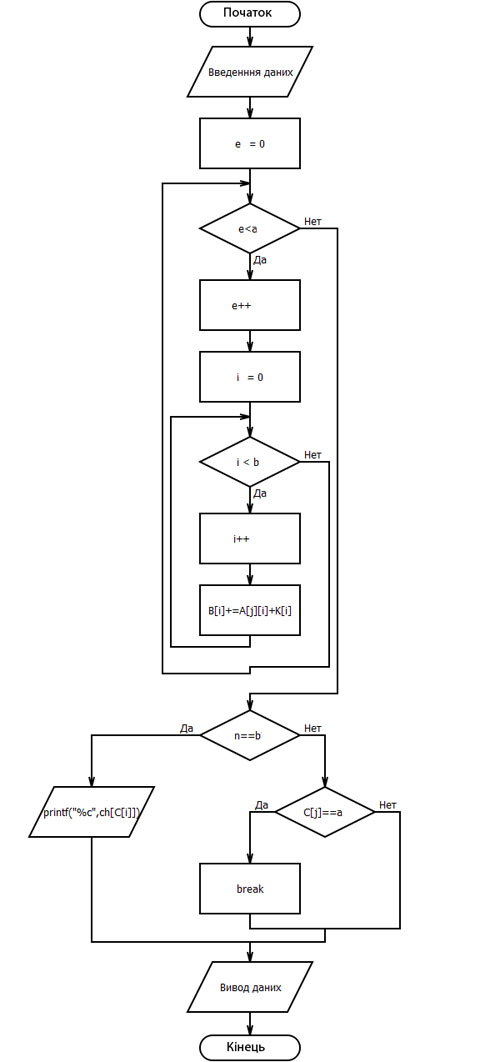
### Мета роботи: набути навичок побудови стовбура графа.

### Порядок виконання роботи:

1. Проаналізувати метод побудови стовбура графа на конкретному прикладі згідно з індивідуальним завданням.
2. Розробити схему алгоритму побудови стовбура графа.
3. Розробити програму, яка реалізує даний алгоритм.
4. Для заданого варіанту привести результати тестування розробленої програми.
5. Розробити інструкцію користувача.
6. Оформити звіт і зробити висновки за результатами роботи.

**Завдання №21**



Блок – схема програми що реалізує побудову стовбура графа.

Виведіть кількість поверхонь

Виведіть поверхню з найменшою кількістю повтореня

Виведіть поверхні з найменшою ціною

Кінець

Рисунок 1. Блок – схема програми що реалізує побудову стовбура графа.

Результати виконання програми.

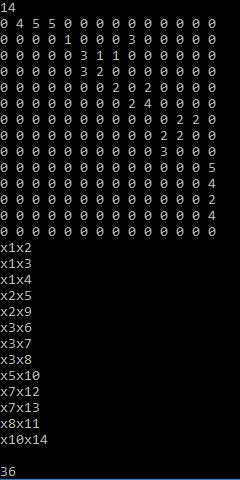


Рисунок 2. Результати виконання програми

*Висновок*: В ході виконання лабораторної роботи було набуто навичок побудови стовбура графа, та було представлено результати тестування програми яку розроблено на мові С++.

### Додаток 2 (лістинг програми, що реалізує побудову стовбура графа).

#include <conio.h>

#include <iostream>

using namespace std;

#define VERTEXES 15

int v;

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL,"Ukrainian");

char arr[13]={'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10','x11','x12','x13'};

int infinity=1000;

int p= VERTEXES;

int a[VERTEXES][VERTEXES];/\*={{ 0, 4, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},

{0,0,0 ,0, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0},

{0, 0, 0, 0, 0, 3, 1, 1, 0, 0, 0, 0, 0, 0},

{0, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0, 0, 0, 0},

{0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0},

{0, 0, 0, 0, 0, 0, 0, 0, 2, 4, 0, 0, 0, 0},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5},

{0, 0 ,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0, 0, 4},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}};\*/

int re;

cin>>re;

for(int n=0;n<13;n++)

{

for(int m=0;m<13;m++)

cin>>a[n][m];

}

int i,j,k={0},g=0;

for(i=0;i<13;i++){

for(j=0;j<13;j++){

for(int l=0;l<13;l++){ a[l][k]=0;}

if(a[i][j]>0){k=j;cout<<"x"<<i+1<<"x"<<j+1<<"\n"; g+=a[i][j];}

}}; cout<<"\n"<<g;

getch();}

Додаток 1 (Інструкція користувача)

1. Запустити програму Stov.exe
2. Ввести кількість вершин
3. Ввести граф представлений у вигляді матриці
4. Натиснупи Enter