

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ
УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Розрахункова робота

з дисципліни «Дискретна математика»

Виконала:

студент групи КН-114

Серкіз Людмила

Викладач:

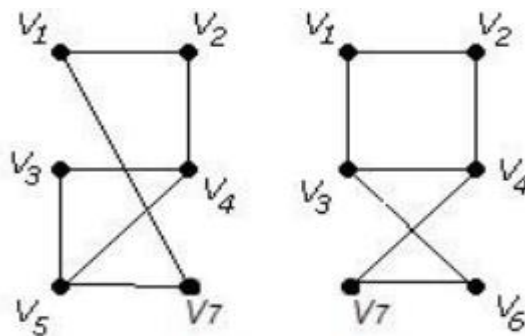
Мельникова Н.І.

Львів – 2019 р.

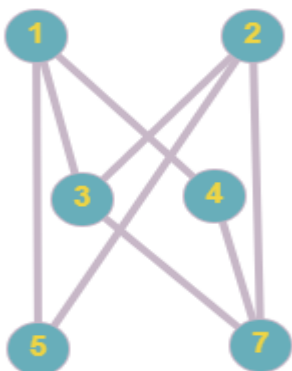
Варіант 19

Завдання № 1

Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об'єднання графів, 3) кільцеву сумму G_1 та G_2 (G_1+G_2), 4) розмножити вершину у другому графі, 5) виділити підграф A - що складається з 3-х вершин в G_1 6) добуток графів.



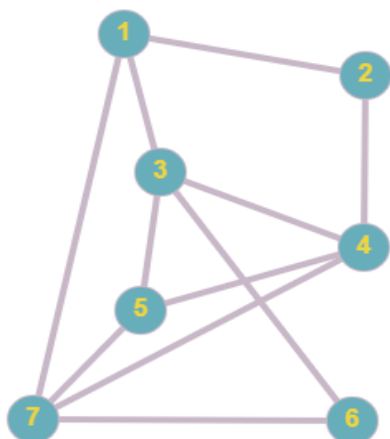
1)Доповнення до першого графу



$$X=\{V_1, V_2, V_3, V_4, V_5, V_7\}$$

$$W=\{(V_1V_3)(V_1V_4)(V_2V_3)(V_2V_5)(V_2V_7)(V_3V_7)(V_4V_7)(V_1V_5)\}$$

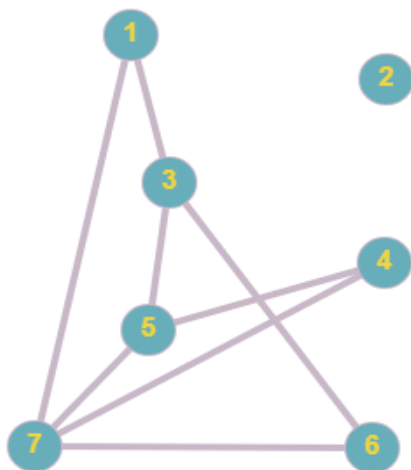
2)Об'єднання графів



$$X=\{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$$

$$W=\{(V_1V_2)(V_1V_3)(V_1V_7)(V_2V_4)(V_3V_4)(V_3V_5)(V_4V_5)(V_5V_7)(V_3V_6)(V_4V_7)(V_6V_7)\}$$

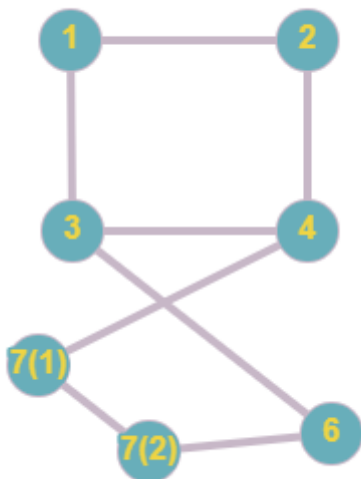
3) Кільцева сума графів



$$X = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$$

$$W = \{(V_1V_3)(V_1V_7)(V_3V_5)(V_3V_6)(V_4V_5)(V_5V_7)(V_4V_7)(V_6V_7)\}$$

4) Розщеплення вершини V_7 на $V_{7(1)}$ та $V_{7(2)}$



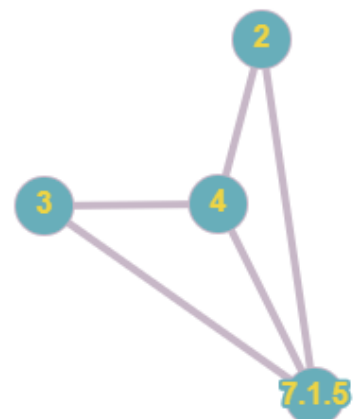
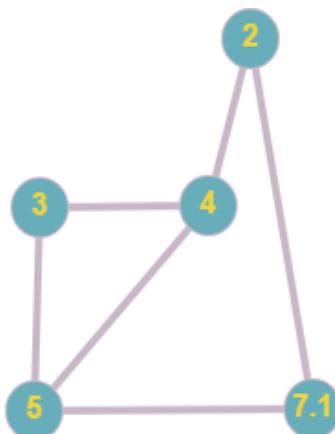
$$X = \{V_1, V_2, V_3, V_4, V_5, V_6, V_{7(1)}, V_{7(2)}\}$$

$$W = \{(V_1V_2)(V_1V_3)(V_2V_4)(V_3V_4)(V_4V_{7(1)})(V_3V_6)(V_6V_{7(2)})(V_{7(1)}V_{7(2)})\}$$

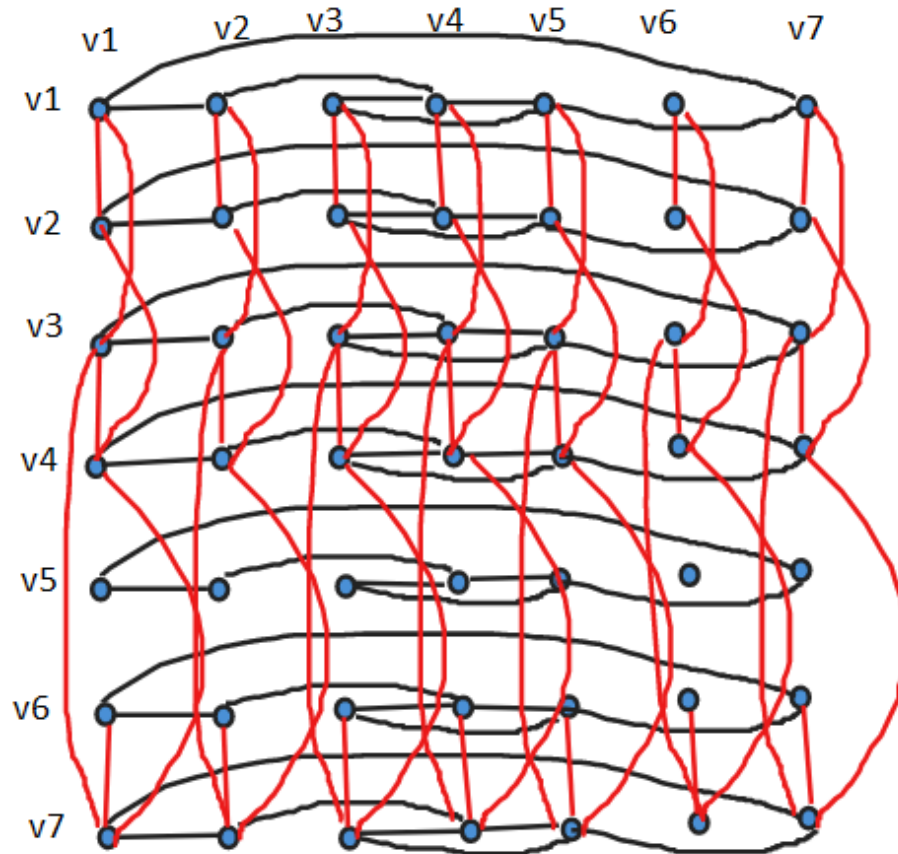
5) 1. Підграф A

2. Стягуємо V_1 до V_2

3. Стягуємо V_5 до V_7V_1

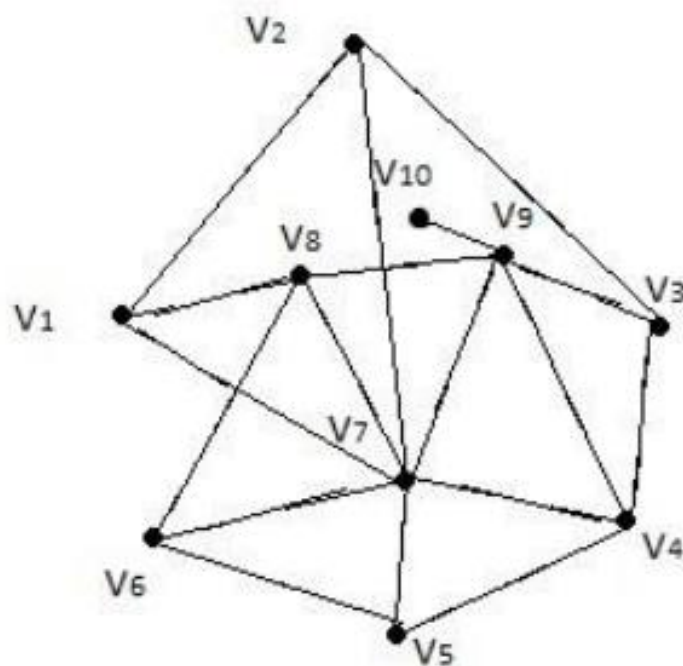


6) Множення графів



Завдання № 2

Скласти таблицю суміжності для орграфа.



Таблиця суміжності

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1	0	1	0	0	0	0	1	1	0	0
V2	1	0	1	0	0	0	1	0	0	0
V3	0	1	0	1	0	0	0	0	1	0
V4	0	0	1	0	1	0	1	0	1	0
V5	0	0	0	1	0	1	1	0	0	0
V6	0	0	0	0	1	0	1	1	0	0
V7	1	1	0	1	1	1	0	1	1	0
V8	1	0	0	0	0	1	1	0	1	0
V9	0	0	1	1	0	0	1	1	0	1
V10	0	0	0	0	0	0	0	0	1	0

Завдання № 3

Для графа з другого завдання знайти діаметр.

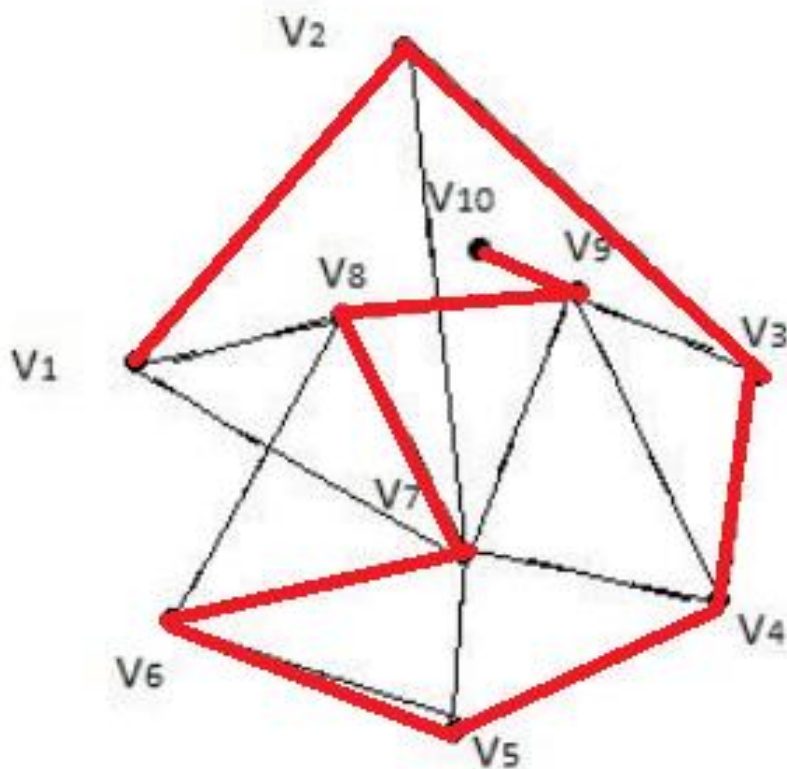
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1	-	1	2	2	2	2	1	1	2	3
V2	1	-	1	2	3	2	1	2	2	3
V3	2	1	-	1	2	3	2	2	1	2
V4	2	2	1	-	1	2	1	2	1	2
V5	2	2	2	1	-	1	1	2	2	3
V6	2	2	3	2	1	-	1	1	2	3
V7	1	1	2	1	1	1	-	1	1	2
V8	1	2	2	2	2	1	1	-	1	2
V9	2	2	1	1	2	2	1	1	-	1
V10	3	3	2	2	3	3	2	2	1	-

Діаметр графа=3

Завдання № 4

Для графа з другого завдання виконати обхід дерева вглиб

Вершина	Номер	Вміст стеку
V1	1	V1
V2	2	V1,V2
V3	3	V1,V2,V3
V4	4	V1,V2,V3,V4
V5	5	V1,V2,V3,V4,V5
V6	6	V1,V2,V3,V4,V5,V6
V7	7	V1,V2,V3,V4,V5,V6,V7
V8	8	V1,V2,V3,V4,V5,V6,V7,V8
V9	9	V1,V2,V3,V4,V5,V6,V7,V8,V9
V10	10	V1,V2,V3,V4,V5,V6,V7,V8,V9,V10
-	-	V1,V2,V3,V4,V5,V6,V7,V8,V9
-	-	V1,V2,V3,V4,V5,V6,V7,V8
-	-	V1,V2,V3,V4,V5,V6,V7
-	-	V1,V2,V3,V4,V5,V6
-	-	V1,V2,V3,V4,V5
-	-	V1,V2,V3,V4
-	-	V1,V2,V3
-	-	V1,V2
-	-	V1
-	-	∅



Програмна перевірка:

```

1  #include<iostream>
2  using namespace std;
3  const int n=10;
4  int i,j;
5  bool *visited=new bool[n];
6  int graph[n][n]={0,1,0,0,0,0,1,1,0,0},
7                  {1,0,1,0,0,0,1,0,0,0},
8                  {0,1,0,1,0,0,0,0,1,0},
9                  {0,0,1,0,1,0,1,0,1,0},
10                 {0,0,0,1,0,1,1,0,0,0},
11                 {0,0,0,0,1,0,1,1,0,0},
12                 {1,1,0,1,1,1,0,1,1,0},
13                 {1,0,0,0,0,1,1,0,1,0},
14                 {0,0,1,1,0,0,1,1,0,1},
15                 {0,0,0,0,0,0,0,0,1,0}};
16 void dfs(int s){
17     int r;
18     cout<<s+1<<" ";
19     visited[s]=true;
20     for(r=0;r<=n;r++)
21         if((graph[s][r]!=0)&&(!visited[r])) dfs(r);
22 }
23 void main( ){
24     int st;
25     cout<<"Matritsa symizhnosti:"<<endl;
26     for(i=0;i<n;i++){
27         visited[i]=false;
28         for(j=0;j<n;j++) cout<<" "<<graph[i][j];
29         cout<<endl;

```

```

30     } cout<<"start=>";
31     cin>>st;
32     bool*vis=new bool[n];
33     cout<<"obxid:";
34     dfs(st-1);
35     delete[]visited;
36 }

```

Matritsa symizhnosti:

```

0 1 0 0 0 0 1 1 0 0
1 0 1 0 0 0 1 0 0 0
0 1 0 1 0 0 0 0 1 0
0 0 1 0 1 0 1 0 1 0
0 0 0 1 0 1 1 0 0 0
0 0 0 0 1 0 1 1 0 0
1 1 0 1 1 1 0 1 1 0
1 0 0 0 0 1 1 0 1 0
0 0 1 1 0 0 1 1 0 1
0 0 0 0 0 0 0 0 1 0

```

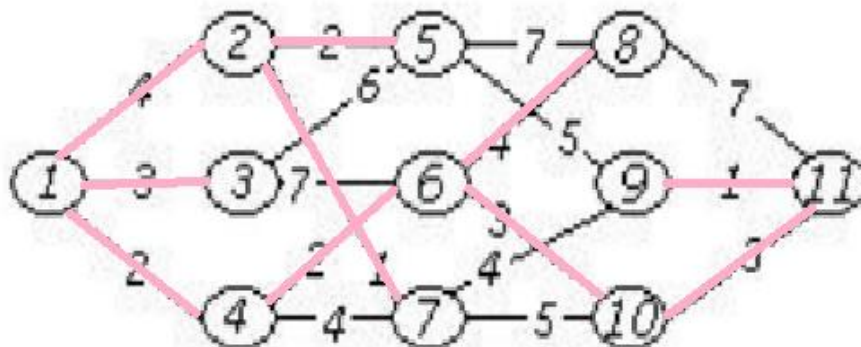
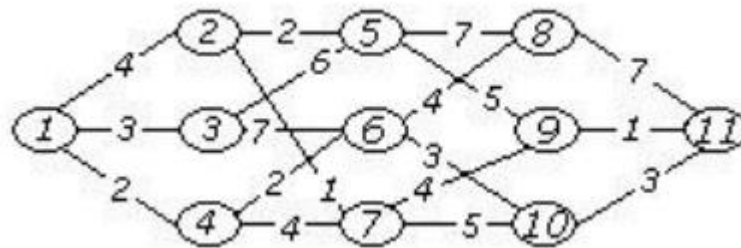
start=>1

obxid:1 2 3 4 5 6 7 8 9 10

...Program finished with exit code 0
Press ENTER to exit console.

Завдання № 5

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



Вага мінімального остового дерева=2+2+3+3+3+1+4+1+2+4=25

Програмна перевірка:

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  { int n,n0;
5    cout<<"Enter the number of vertices: ";
6    cin>>n;
7    cout<<"\nEnter the number of edges: " << endl;
8    cin>>n0;
9    cout<<"\nnow\nenter the data in order: \nA(1) A(2) A(1)A(2)(weight of edges)\n";
10   int ** arr;
11   arr=new int*[n] ;
12   for(int i=0;i<n;i++){
13     arr[i]=new int[n];
14     for(int j=0;j<n;j++)
15     { arr[i][j]=0;}
16   }
17   int a1,a2,w;
18   for(int i=0;i<n0;i++){
19     cin>> a1 >> a2 >> w;
20     arr[a1-1][a2-1]=w;
21     arr[a2-1][a1-1]=w;
22   }
23   int* mas;
24   mas=new int[n];
25   for(int i=0;i<n;i++){
26     mas[i]=0;}
27   mas[0]=1;
28   int sum=0;
29   cout<<"\n\n";
30   for(int r=0,t=0;r!=(n-1);)
31   { t=0;
32     for(int i=0;i<n;i++){
33       if(mas[i]!=0){
34         for(int j=0;j<n;j++){
35           if(arr[i][j]!=0){
36             if(t==0){
37               a1=i;
38               a2=j;
39               w=arr[i][j];
40               t++;
41             }
42             else{
43               if(arr[i][j]<w){
44                 a1=i;
45                 a2=j;
46                 w=arr[i][j];
47               }
48             }
49           }
50         }
51       }
52     }
53     sum+=w;
54     cout<<"\nA("<<a1+1<<")->A("<<a2+1<<")="<<w;
55     mas[a2]=a2+1;
56     arr[a1][a2]=0;
57     arr[a2][a1]=0;
58     r=0;
```

```

58     }
59     for(int q=0;q<n;q++){
60         if(mas[q]!=0){
61             r++;}
62     }
63 }
64 for(int i=0;i<n;i++){
65     if(mas[i]==0){a2=i;}
66 } for(int i=0,t=0;i<n;i++){
67     if(arr[i][a2]!=0){
68         if(t==0){
69             a1=i;
70             w=arr[i][a2];
71             t++;
72         }
73         else{
74             if(arr[i][a2]<w){
75                 a1=i;
76                 w=arr[i][a2];
77             }
78         }
79     }
80 }
81 sum+=w;
82 cout<<"\nA("<<a1+1<<")->A("<<a2+1<<")="<<w;
83 cout<<"\n\nminimal weight:"<<sum;
84 cout<<"\n\n\n";
85 return 0;
86 }

```

Результати:

```

Enter the number of vertices: 11
Enter the number of edges:
18
now
enter the data in order:
A(1) A(2) A(1)A(2) (weight of edges)
1 2 4
1 3 3
1 4 2
2 5 2
2 7 1
3 5 6
3 6 7
4 6 2
4 7 4
5 8 7
5 9 5
6 8 4
6 10 3
7 9 4
7 10 5
8 11 7
9 11 1
10 11 3

```

```

A(1)->A(4)=2
A(4)->A(6)=2
A(1)->A(3)=3
A(6)->A(10)=3
A(10)->A(11)=3
A(11)->A(9)=1
A(1)->A(2)=4
A(2)->A(7)=1
A(2)->A(5)=2
A(6)->A(8)=4

```

minimal weight:25

...Program finished with exit code 0
Press ENTER to exit console.

Завдання № 6

Розв'язати задачу комівояжера для повного 8-ми вершинного графа методом «іди у найближчий», матриця вагів якого має вигляд:

	1	2	3	4	5	6	7	8
1	∞	2	2	2	2	3	2	2
2	2	∞	5	1	2	3	2	4
3	2	5	∞	6	6	5	1	5
4	2	1	6	∞	6	6	6	6
5	2	2	6	6	∞	5	1	5
6	3	3	5	6	5	∞	2	1
7	2	2	1	6	1	2	∞	5
8	2	4	5	6	5	1	5	∞

Варіант розв'язування 1

	1	3	2,4	5	6	7	8
1	∞	2	2	2	3	2	2
3	2	∞	6	6	5	1	5
2,4	2	6	∞	6	6	6	6
5	2	6	6	∞	5	1	5
6	3	5	6	5	∞	2	1
7	2	1	6	1	2	∞	5
8	2	5	6	5	1	5	∞

	3	2,4,1	5	6	7	8
3	∞	6	6	5	1	5
2,4,1	6	∞	6	6	6	6
5	6	6	∞	5	1	5
6	5	6	5	∞	2	1
7	1	6	1	2	∞	5
8	5	6	5	1	5	∞

	2,4,1,3	5	6	7	8
2,4,1,3	∞	6	6	6	6
5	6	∞	5	1	5
6	6	5	∞	2	1
7	6	1	2	∞	5
8	6	5	1	5	∞

	2,4,1,3,5	6	7	8
2,4,1,3,5	∞	6	6	6
6	6	∞	2	1
7	6	2	∞	5
8	6	1	5	∞

	2,4,1,3,5,6	7	8
2,4,1,3,5,6	∞	6	6
7	6	∞	5
8	6	5	∞

	2,4,1,3,5,6,7	8
2,4,1,3,5,6,7	∞	6
8	6	∞

Варіант2

	1	2	3	4	5	6	7	8
1	∞	2	2	2	2	3	2	2
2	2	∞	5	1	2	3	2	4
3	2	5	∞	6	6	5	1	5
4	2	1	6	∞	6	6	6	6
5	2	2	6	6	∞	5	1	5
6	3	3	5	6	5	∞	2	1
7	2	2	1	6	1	2	∞	5
8	2	4	5	6	5	1	5	∞

	1	2	4	5	6	3,7	8
1	∞	2	2	2	3	2	2
2	2	∞	1	2	3	2	4
4	2	1	∞	6	6	6	6
5	2	2	6	∞	5	1	5
6	3	3	6	5	∞	2	1
3,7	2	2	6	1	2	∞	5
8	2	4	6	5	1	5	∞

	1	2	4	6	3,7,5	8
1	∞	2	2	3	2	2
2	2	∞	1	3	2	4
4	2	1	∞	6	6	6
6	3	3	6	∞	2	1
3,7,5	2	2	6	2	∞	5
8	2	4	6	1	5	∞

	2	4	6	3,7,5,1	8
2	∞	1	3	2	4
4	1	∞	6	6	6
6	3	6	∞	2	1
3,7,5,1	2	6	2	∞	5
8	4	6	1	5	∞

	4	6	3,7,5,1,2	8
4	∞	6	6	6
6	6	∞	2	1
3,7,5,1,2	6	2	∞	5
8	6	1	5	∞

	4	3,7,5,1,2,6	8
4	∞	6	6
3,7,5,1,2,6	6	∞	5
8	6	5	∞

	4	3,7,5,1,2,6,8
4	∞	6
3,7,5,1,2,6,8	6	∞

Варіант 3

	1	2	3	4	5	7	6,8
1	∞	2	2	2	2	2	2
2	2	∞	5	1	2	2	4
3	2	5	∞	6	6	1	5
4	2	1	6	∞	6	6	6
5	2	2	6	6	∞	1	5
7	2	2	1	6	1	∞	5
6,8	2	4	5	6	5	5	∞

	2	3	4	5	7	6,8,1
2	∞	5	1	2	2	4
3	5	∞	6	6	1	5
4	1	6	∞	6	6	6
5	2	6	6	∞	1	5
7	2	1	6	1	∞	5
6,8,1	4	5	6	5	5	∞

	3	4	5	7	6,8,1,2
3	∞	6	6	1	5
4	6	∞	6	6	6
5	6	6	∞	1	5
7	1	6	1	∞	5
6,8,1,2	5	6	5	5	∞

	4	5	7	6,8,1,2,3
4	∞	6	6	6
5	6	∞	1	5
7	6	1	∞	5
6,8,1,2,3	6	5	5	∞

	4	7	6,8,1,2,3,5
4	∞	6	6
7	6	∞	5
6,8,1,2,3,5	6	5	∞

	4	6,8,1,2,3,5,7
4	∞	6
6,8,1,2,3,5,7	6	∞

Програмна перевірка:

```

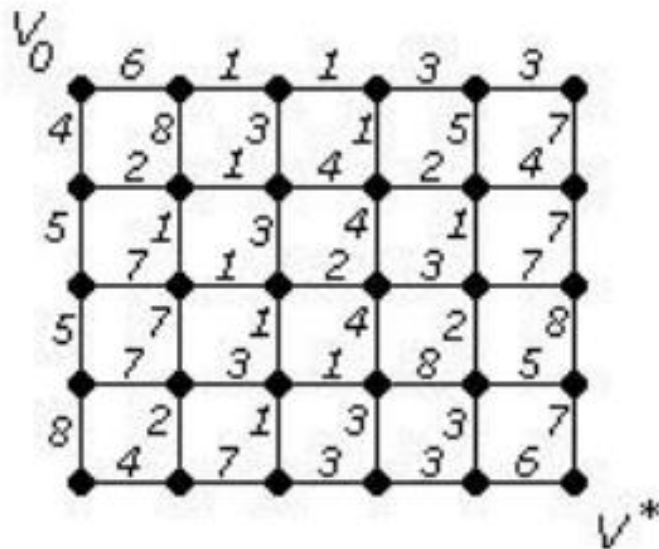
1 #include<iostream>
2 #define inf 1E9
3 #define NMAX 8
4 #define min(x,y) x<y ? x:y
5 using namespace std;
6 int m,k,n,j,i ,temp,ans,d[NMAX][NMAX],t[1<<NMAX][NMAX];
7 int get(int nmb,int x){ return(x&(1<<nmb))!=0;}
8 int main(){
9     cin>>n;
10    for (i=0;i<n;++i)
11        for (j=0;j<n;++j)
12        cin>>d[i][j];
13    t[1][0]=0; m=1<<n;
14    for(i=1;i<m;i+=2)
15        for(j=(i==1)?1:0;j<n;++j){
16            t[i][j]=inf;
17            if(j>0&&get(j,i)){
18                temp=i^(1<<j);
19                for(k=0;k<n;++k)
20                    if(get(k,i)&&d[k][j]>0) t[i][j]=min(t[i][j],t[temp][k]+d[k][j]);
21            }
22        }
23    }for(j=1,ans=inf;j<n;++j)
24    if (d[j][0]>0)ans=min(ans,t[m-1][j]+d[j][0]);
25    if(ans==inf) cout<<-1;
26    else cout <<ans;
27 }

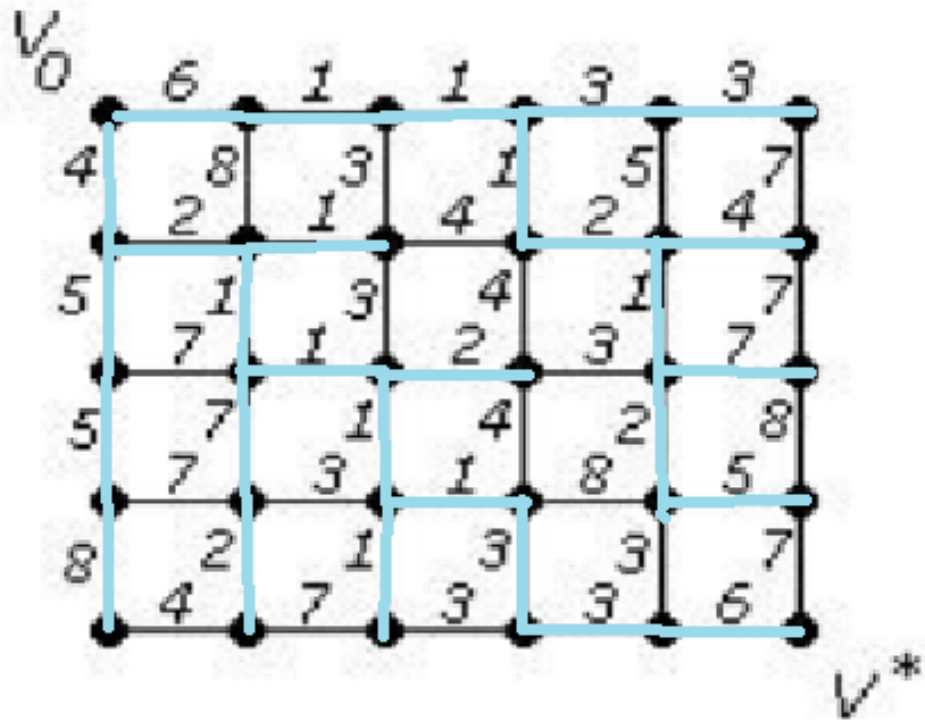
```

Starting program: /home/a.ou
8
1 2 2 2 2 3 2 2
2 1 5 1 2 3 2 4
2 5 1 6 6 5 1 5
2 1 6 1 6 6 6 6
2 2 6 6 1 5 1 5
3 3 5 6 5 1 2 1
2 2 1 6 1 2 1 5
2 4 5 6 5 1 5 1
6 [Inferior 1 (process 3222)
(qdb)

Завдання № 7

За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі між парою вершин V_0 і V^* .





$$S(V_0, V^*) = 4 + 2 + 1 + 1 + 1 + 1 + 3 + 3 + 6 = 22$$

Програмна перевірка:

```

1  #include<iostream>
2  using namespace std;
3  int n;
4  int i, j, q;
5  int dist[40];
6  bool visited[40];
7  int pred[40];
8  void createGraph(int c[40][40])
9  {
10     int g1, g2;
11     cout << "Enter the number of vertices: ";
12     cin >> n;
13     for (int i = 0; i < n; i++) {
14         for (int j = 0; j < n; j++) {
15             c[i][j] = 0;
16         }
17     }
18     cout << "Enter size of (n*m) : ";
19     cin >> g1 >> g2;
20     for (i = 0; i < n; i++) {
21         for (j = i + 1; j < n; j++) {
22             if (j == i + 1 || j == i + g1) {
23                 cout << "Enter the length from x" << i+1 << " to x" << j+1 << ": ";
24                 cin >> c[i][j];
25             } else {c[i][j] = 0;}}
26     }
27     int minDistance()
28     {
29         int minimum = 10000, minDist;
30         for (int v = 0; v < n; v++)
31             if (visited[v] == false && dist[v] <= minimum)
32                 {minimum = dist[v];

```

```

30     minDist = v;}
31     return minDist;}
32 void printPath(int j)
33 {if (pred[j] == -1)
34     return;
35     printPath(pred[j]);
36     cout << "X" << j+1 << " -> ";}
37 void dijkstra(int c[40][40])
38 {
39     int start;
40     cout << "Enter the first node : ";
41     cin >> start;
42     for (int i = 0; i < n; i++) {
43         pred[i] = -1;
44         dist[i] = 10000;
45         visited[i] = false; }
46     dist[start-1] = 0;
47     for (int count = 0; count < n - 1; count++) {
48         int u = minDistance();
49         visited[u] = true;
50         for (int v = 0; v < n; v++)
51             if (!visited[v] && c[u][v] &&
52                 dist[u] + c[u][v] < dist[v]) {
53                 pred[v] = u;
54                 dist[v] = dist[u] + c[u][v];} }
55     cout << "The least way is: ";
56     cout << dist[29] << endl;
57     cout << "The way is: ";
58     cout << "X1 -> ";
59     printPath(29);
60     cout << "The end!") << endl;}
61 int main()
62 {
63     int c[40][40];
64     createGraph(c);
65     dijkstra(c);
66     return 0;
67 }

```

Результати

Enter the number of vertices: 30	Enter the length from x13 to x19: 5
Enter size of (n*m) : 6 5	Enter the length from x14 to x15: 1
Enter the length from x1 to x2: 6	Enter the length from x14 to x20: 7
Enter the length from x1 to x7: 4	Enter the length from x15 to x16: 2
Enter the length from x2 to x3: 1	Enter the length from x15 to x21: 1
Enter the length from x2 to x8: 8	Enter the length from x16 to x17: 3
Enter the length from x3 to x4: 1	Enter the length from x16 to x22: 4
Enter the length from x3 to x9: 3	Enter the length from x17 to x18: 7
Enter the length from x4 to x5: 3	Enter the length from x17 to x23: 2
Enter the length from x4 to x10: 1	Enter the length from x18 to x19: 0
Enter the length from x5 to x6: 3	Enter the length from x18 to x24: 8
Enter the length from x5 to x11: 5	Enter the length from x19 to x20: 7
Enter the length from x6 to x7: 0	Enter the length from x19 to x25: 8
Enter the length from x6 to x12: 7	Enter the length from x20 to x21: 3
Enter the length from x7 to x8: 2	Enter the length from x20 to x26: 2
Enter the length from x7 to x13: 5	Enter the length from x21 to x22: 1
Enter the length from x8 to x9: 1	Enter the length from x21 to x27: 1
Enter the length from x8 to x14: 1	Enter the length from x22 to x23: 8
Enter the length from x9 to x10: 4	Enter the length from x22 to x28: 3
Enter the length from x9 to x15: 3	Enter the length from x23 to x24: 5
Enter the length from x10 to x11: 2	Enter the length from x23 to x29: 3
Enter the length from x10 to x16: 4	Enter the length from x24 to x25: 0
Enter the length from x11 to x12: 4	Enter the length from x24 to x30: 7
Enter the length from x11 to x17: 1	Enter the length from x25 to x26: 4
Enter the length from x12 to x13: 0	Enter the length from x26 to x27: 7
Enter the length from x12 to x18: 7	Enter the length from x27 to x28: 3
Enter the length from x13 to x14: 7	Enter the length from x28 to x29: 3

```

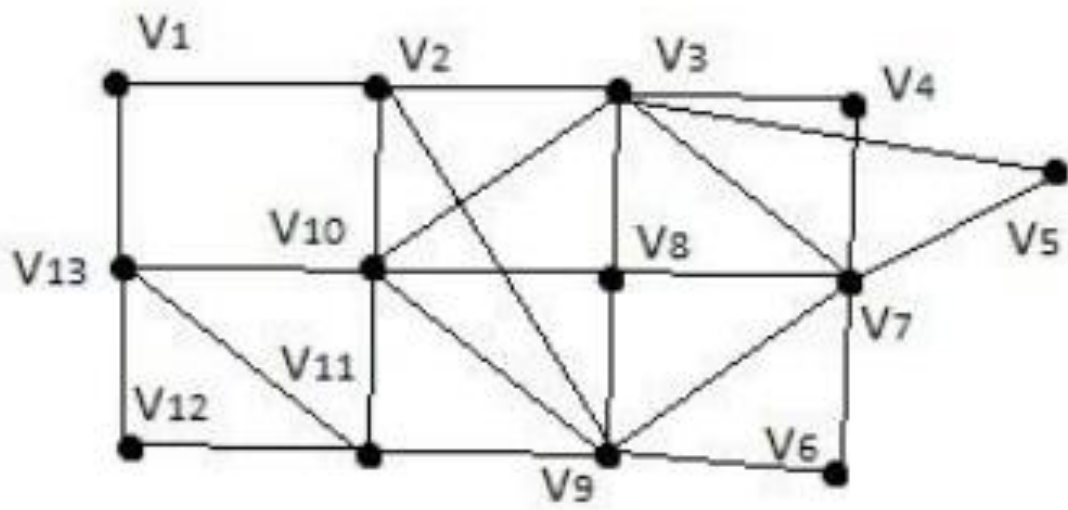
Enter the length from x29 to x30: 6
Enter the first node : 1
The least way is: 22
The way is: X1 -> X7 -> X8 -> X14 -> X15 -> X21 -> X27 -> X28 -> X29 -> X30 -> The end!)

...Program finished with exit code 0
Press ENTER to exit console.

```

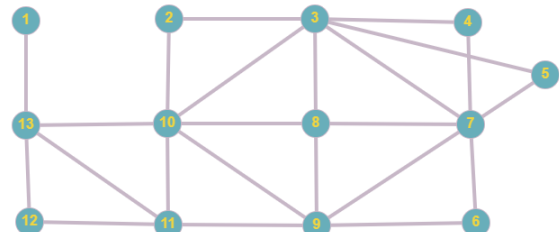
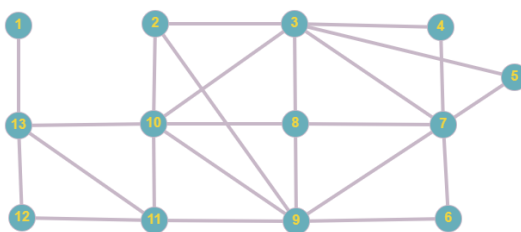
Завдання № 8

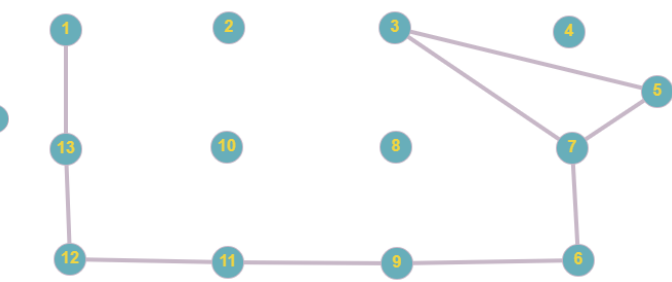
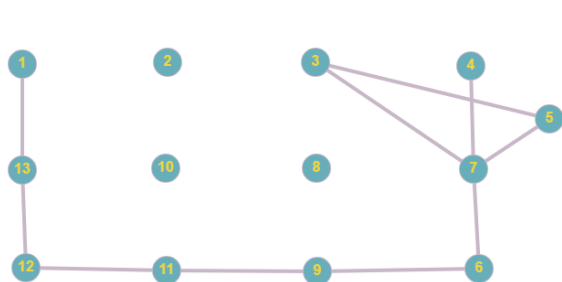
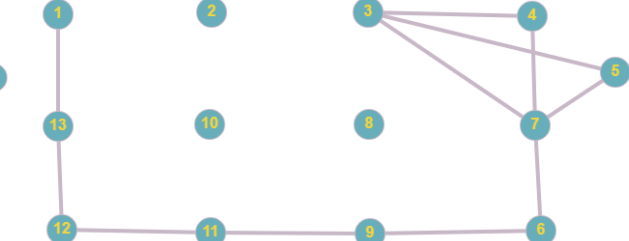
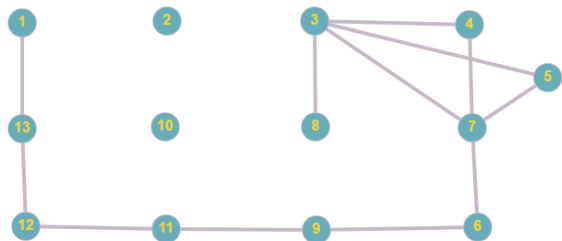
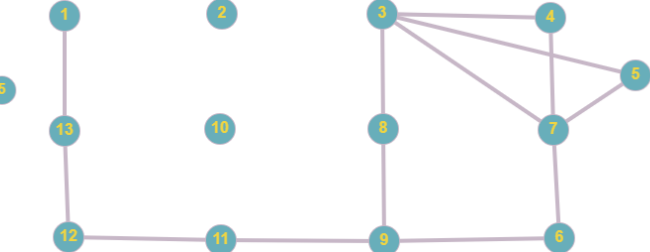
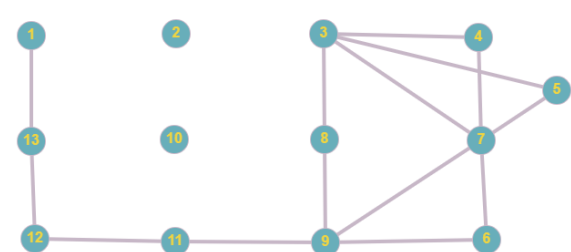
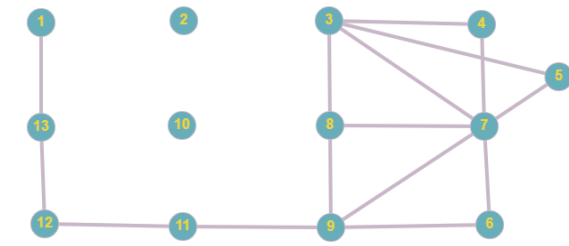
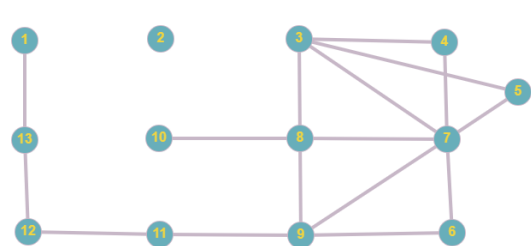
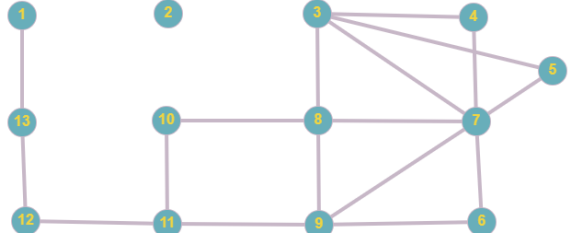
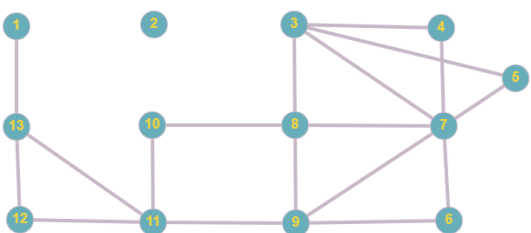
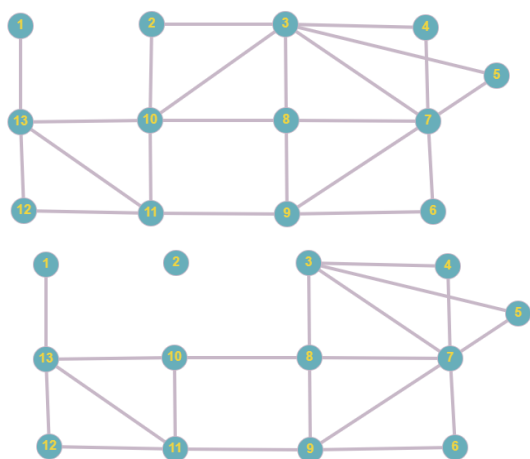
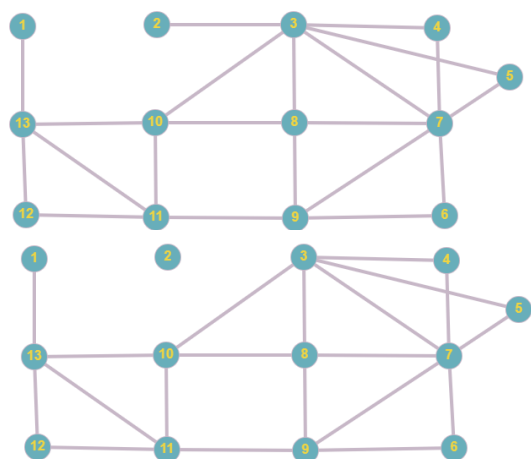
Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.

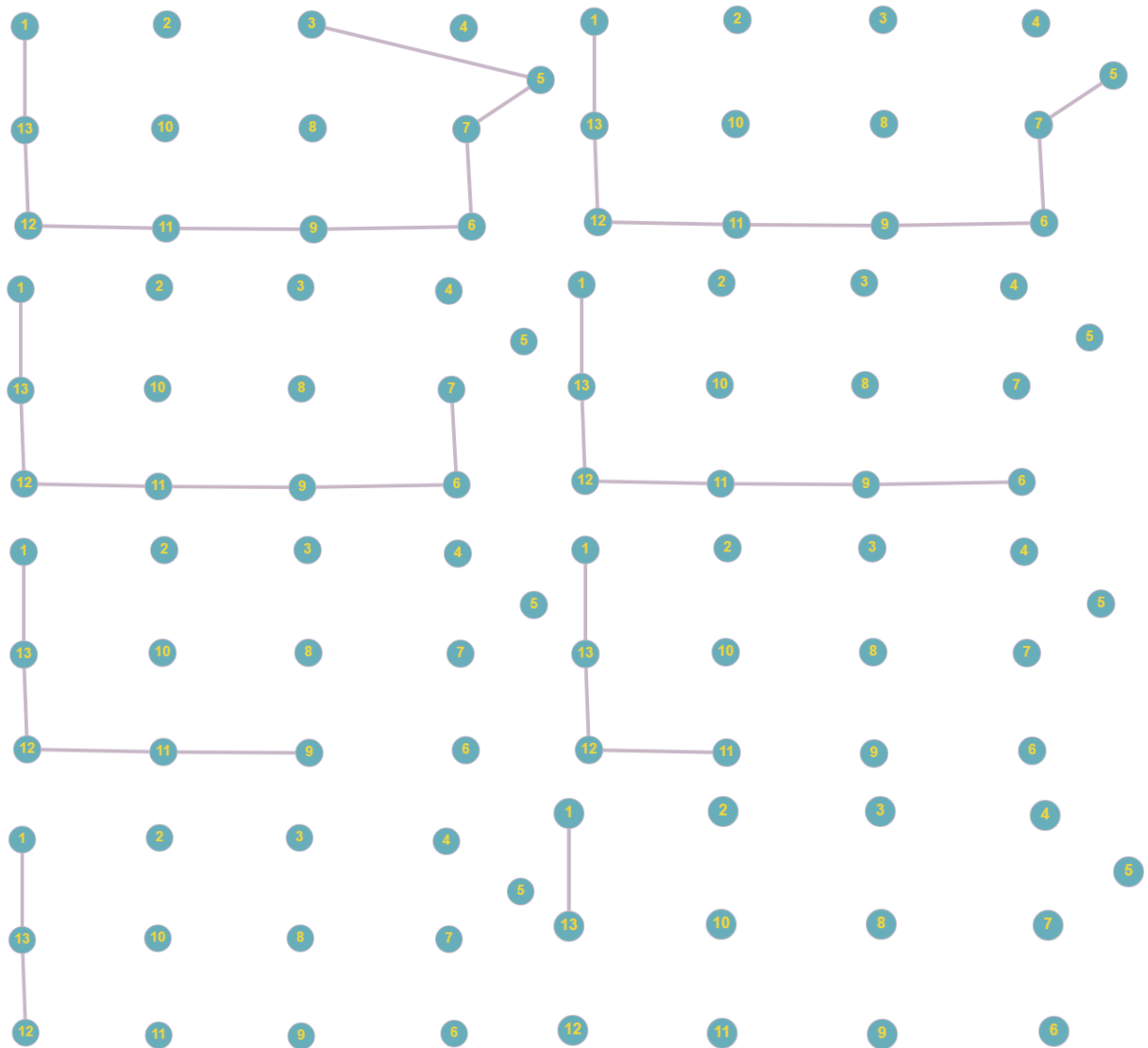


Алгоритм Флері

Починаємо з вершини 1, що належить V і кожен раз викреслюємо пройдене ребро







Програмна перевірка:

```

1  #include<iostream>
2  #define NODE 13
3  using namespace std;
4  int graph[NODE][NODE]={0,1,0,0,0,0,0,0,0,0,0,0,1},
5  {1,0,1,0,0,0,0,0,1,1,0,0,0},
6  {0,1,0,1,1,0,1,1,0,1,0,0,0},
7  {0,0,1,0,0,0,0,1,0,0,0,0,0},
8  {0,0,1,0,0,0,0,1,0,0,0,0,0},
9  {0,0,0,0,0,0,0,1,0,1,0,0,0},
10 {0,0,1,1,1,1,0,0,1,0,0,0,0},
11 {0,0,1,0,0,0,0,0,0,1,1,0,0},
12 {0,1,0,0,0,1,1,1,0,1,1,0,0},
13 {0,1,1,0,0,0,0,0,1,1,0,1,1},
14 {0,0,0,0,0,0,0,0,0,1,1,0,1},
15 {0,0,0,0,0,0,0,0,0,0,1,0,1},
16 {1,0,0,0,0,0,0,0,0,0,1,1,0}};
17 int tg[NODE][NODE];
18 int FSV(){
19     for(int i=0;i<NODE;i++){
20         int deg=0;
21         for(int j=0;j<NODE;j++){
22             if(tg[i][j]) deg++;
23             if (deg%2!=0) return i;
24         }return 0;
25     } bool isb(int u,int v){
26         int deg=0;
27         for (int i=0;i<NODE;i++)
28             if(tg[v][i]) deg++;
29         if (deg>1){return false;}
30     }

```

```

30     return true;}
31 int edc(){
32     int count=0;
33     for(int i=0;i<NODE;i++)
34         for(int j=i;j<NODE;j++)
35             if(tg[i][j]) count++;
36     return count;
37 }void fleri(int s){
38     static int edge=edc();
39     for(int v=0;v<NODE;v++){
40         if(tg[s][v]){
41             if(edge<=1||!isb(s,v)){
42                 cout<<s<<"="<<v<<" ";
43                 tg[s][v]=tg[v][s]=0;
44                 edge--;
45                 fleri(v);
46             }
47         }
48     }
49 }int main(){
50     for(int i=0;i<NODE;i++)
51         for(int j=0;j<NODE;j++)
52             tg[i][j]=graph[i][j];
53     fleri(FSV());
54 }

```

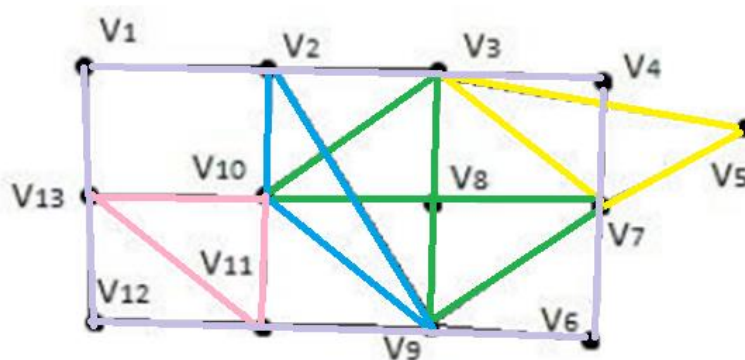
```

6=>2 2=>1 1=>0 0=>12 12=>9 9=>1 1=>8 8=>5 5=>6 6=>3 3=>2 2=>4 4=>6 6=>8 8=>7 7=>2 2=>9 9=>8 8=>10 10=>9 10=>11 11=>
12
...Program finished with exit code 0
Press ENTER to exit console.

```

Алгоритм на основі циклів

Ейлерів цикл - це об'єднання всіх простих циклів графа.



Завдання №9

Спростити формули (привести їх до скороченої ДНФ).

$$19. \overline{\overline{xy(x\bar{y}z \vee \bar{x}y)}}$$

1) \overline{xy} 2) \overline{xyz} 3) \overline{xy} 4) $\overline{xyz \vee xy}$ 5) $\overline{xy(xyz \vee xy)}$

x	y	z	$\neg y$	$\neg x$	1	2	3	4	5	6
0	0	0	1	1	1	0	0	0	0	1
0	0	1	1	1	1	0	0	0	0	1
0	1	0	0	1	1	0	1	1	1	0
1	0	0	1	0	1	0	0	0	0	1
1	0	1	1	0	1	1	0	1	1	0
0	1	1	0	1	1	0	1	1	1	0
1	1	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	1

Карта Карно:

X\YZ	00	01	11	10
0	1	1	0	0
1	1	0	1	1

Мінімізуємо ф-цію:

$$\overline{\overline{xy}} \vee \overline{xy} \vee \overline{xz}$$