

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Розрахункова робота**

з дисципліни

«Дискретна математика»

**Виконала:**

студентка групи КН-114

Кемська Юлія

**Викладач:**

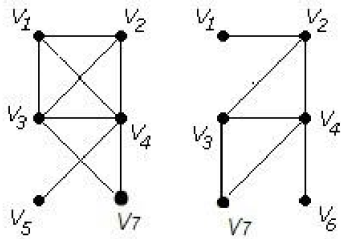
Мельникова Н.І.

Львів – 2019

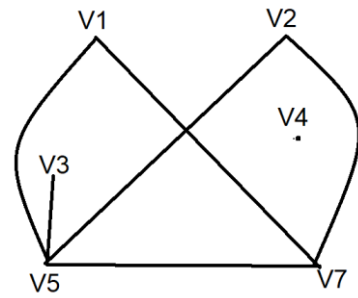
## Варіант- 15

Завдання № 1 Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об'єднання графів, 3) кільцеву суму  $G1$  та  $G2$  ( $G1+G2$ ), 4) розмножити вершину у другому графі, 5) виділити підграф  $A$  - що складається з 3-х вершин в  $G1$  6) добуток графів.

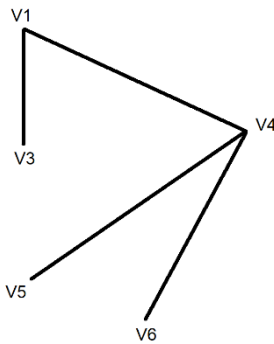
15)



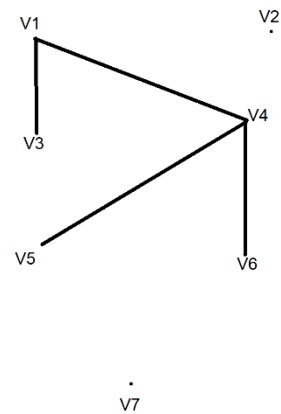
1)



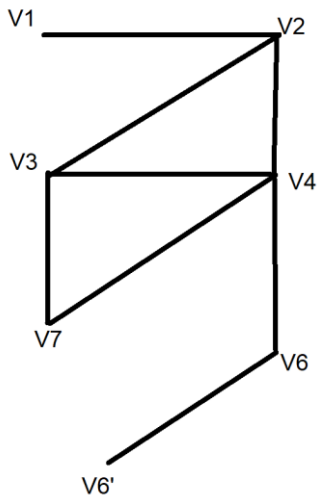
2)



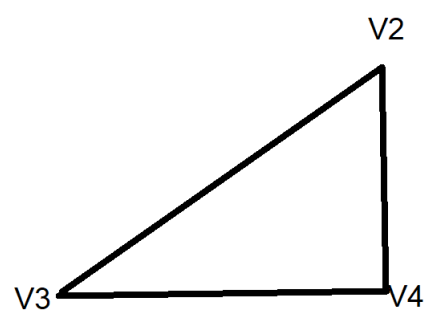
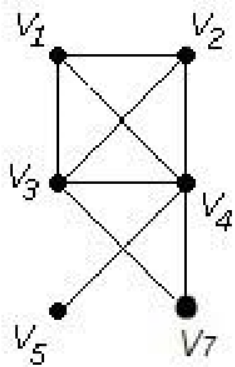
3)



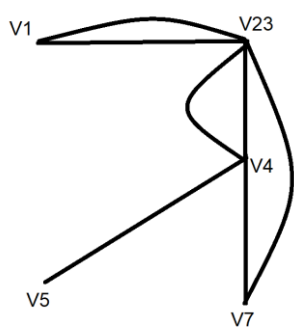
4)



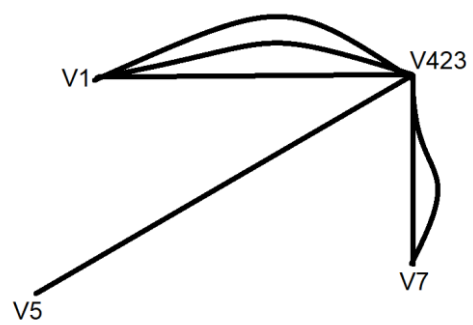
5)



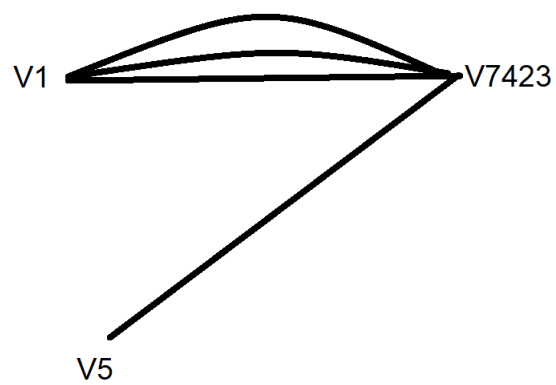
1)



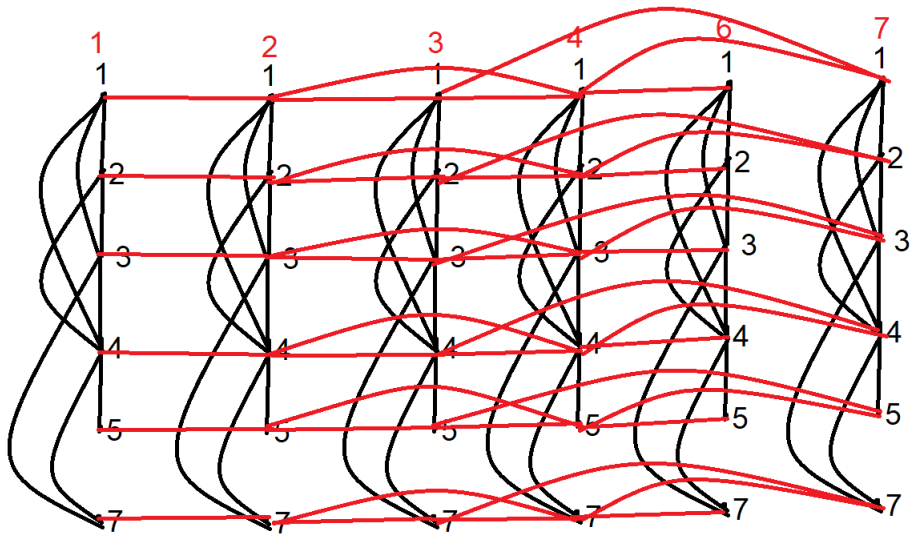
2)



3)

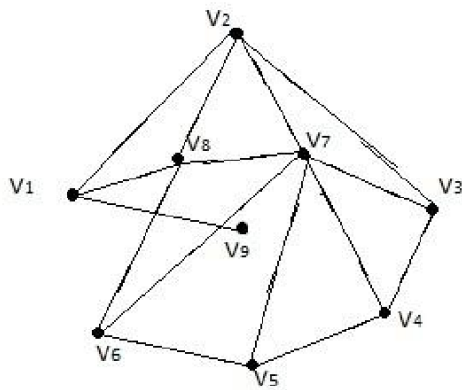


6)



Завдання № 2 Скласти таблицю суміжності для орграфа.

15)



	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	0	1	1
2	1	0	1	0	0	0	1	1	0
3	0	1	0	1	0	0	1	0	0
4	0	0	1	0	1	0	1	0	0
5	0	0	0	1	0	1	1	0	0
6	0	0	0	0	1	0	1	1	0
7	0	1	1	1	1	1	0	1	0
8	1	1	0	0	0	1	1	0	0
9	1	0	0	0	0	0	0	0	0

Завдання № 3 Для графа з другого завдання знайти діаметр.

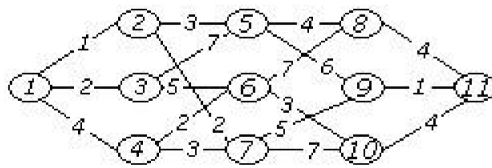
D = 4

Завдання № 4 Для графа з другого завдання виконати обхід дерева вглиб.

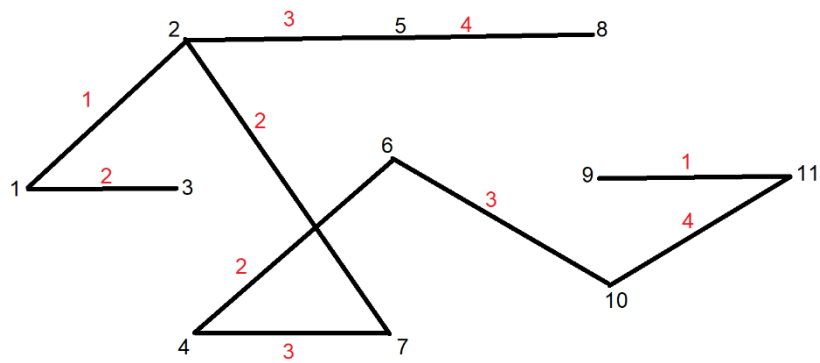
1	1
1 2	2
1 8 8	8
1 2 8 3	3
1 2 8 3 4	4
1 2 8 3 4 5	5
1 2 8 3 4 5 6	6
1 2 8 3 4 5 6 7	7
1 2 8 3 4 5 6	-
1 2 8 3 4 5	-
1 2 8 3 4	-
1 2 8 3	-
1 2 8	-
1 2	-
1	-
1 9	9
1	-
0	-

Завдання № 5 Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

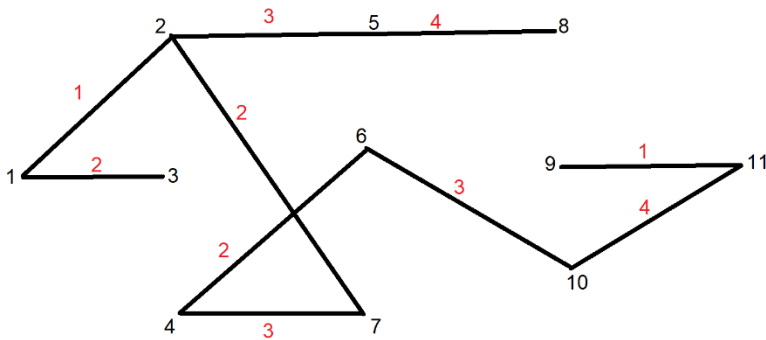
15)



**Краскала**



**Прима**



Завдання № 6 Розв'язати задачу комівояжера для повного 8-ми вершинного графа методом «іди у найближчий», матриця вагів якого має вигляд:

15)

	1	2	3	4	5	6	7	8
1	$\infty$	3	2	1	2	2	3	2
2	3	$\infty$	6	5	4	5	1	2
3	2	6	$\infty$	3	2	1	3	3
4	1	5	3	$\infty$	5	1	5	1
5	2	4	2	5	$\infty$	2	2	2
6	2	5	1	1	2	$\infty$	7	5
7	3	1	3	5	2	7	$\infty$	5
8	2	2	3	1	2	5	5	$\infty$

	1	2	3	4	5	6	7	8
1	$\infty$	3	2	1	2	2	3	2
2	3	$\infty$	6	5	4	5	1	2
3	2	6	$\infty$	3	2	1	3	3
4	1	5	3	$\infty$	5	1	5	1
5	2	4	2	5	$\infty$	2	2	2
6	2	5	1	1	2	$\infty$	7	5
7	3	1	3	5	2	7	$\infty$	5
8	2	2	3	1	2	5	5	$\infty$

	2	3	4	5	6	7	8
2	$\infty$	6	5	4	5	1	2
3	6	$\infty$	3	2	1	3	3
4	5	3	$\infty$	5	1	5	1
5	4	2	5	$\infty$	2	2	2
6	5	1	1	2	$\infty$	7	5
7	1	3	5	2	7	$\infty$	5
8	2	3	1	2	5	5	$\infty$



	2	3	5	6	7	8
2	$\infty$	6	4	5	1	2
3	6	$\infty$	2	1	3	3
5	4	2	$\infty$	2	2	2
6	5	1	2	$\infty$	7	5
7	1	3	2	7	$\infty$	5
8	2	3	2	5	5	$\infty$

	2	3	5	7	8
2	$\infty$	6	4	1	2
3	6	$\infty$	2	3	3
5	4	2	$\infty$	2	2
7	1	3	2	$\infty$	5
8	2	3	2	5	$\infty$

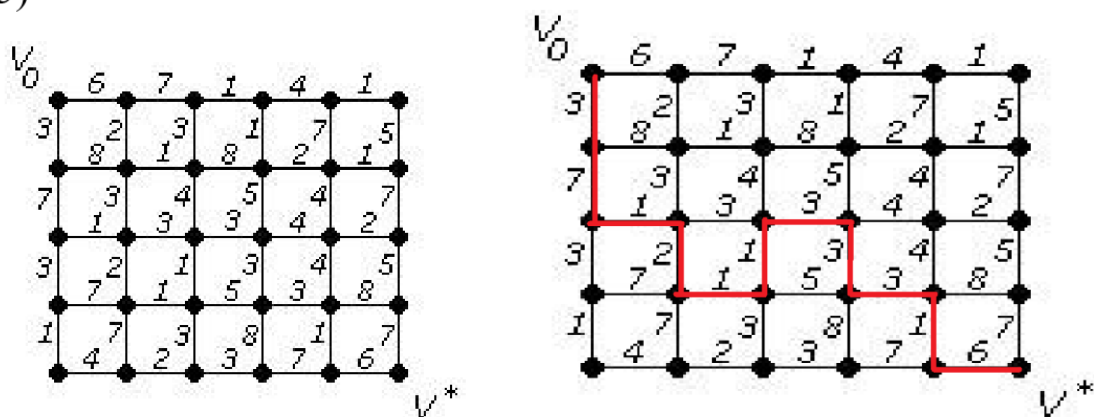
	2	5	7	8
2	$\infty$	4	1	2
5	4	$\infty$	2	2
7	1	2	$\infty$	5
8	2	2	5	$\infty$

	2	7	8
2	$\infty$	1	2
7	1	$\infty$	5
8	2	5	$\infty$

	2	8
2	$\infty$	2
8	2	$\infty$

Завдання № 7 За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі між парою вершин  $V_0$  і  $V^*$ .

15)

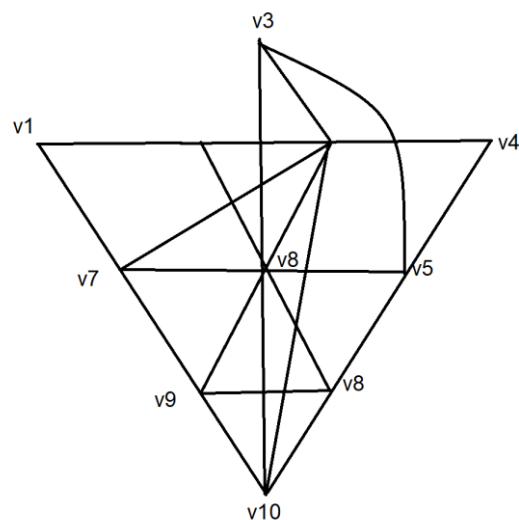
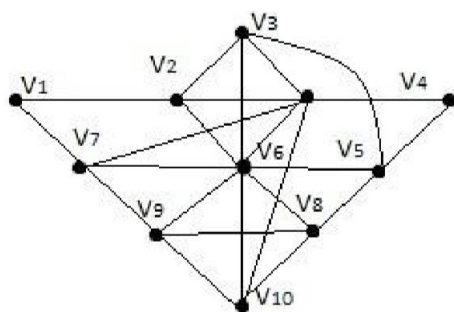


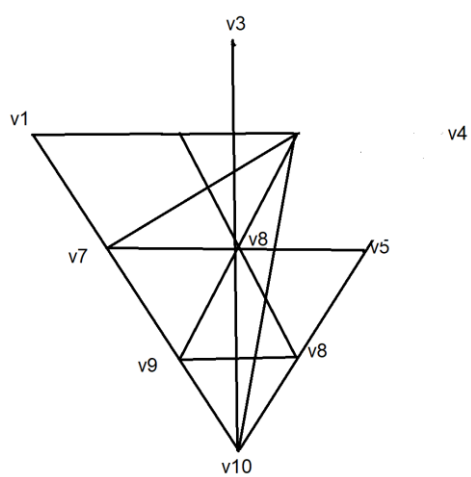
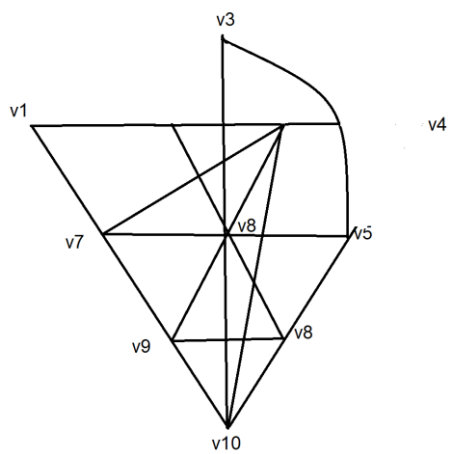
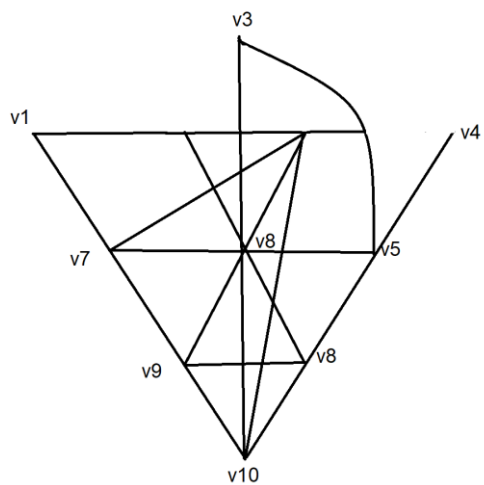
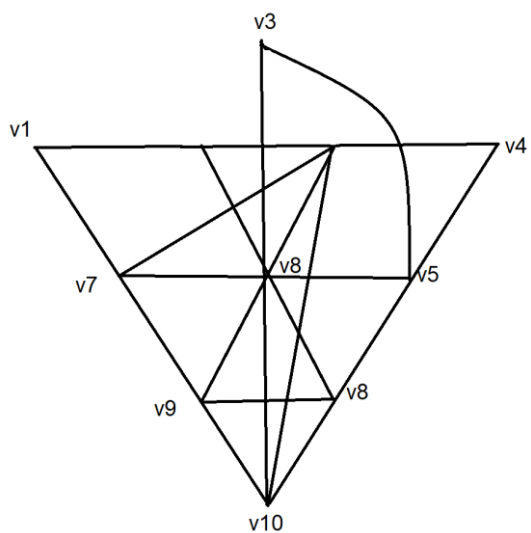
Найменший шлях: **31**

$$3+7+1+2+1+1+3+3+3+1+6=31$$

## Завдання № 8

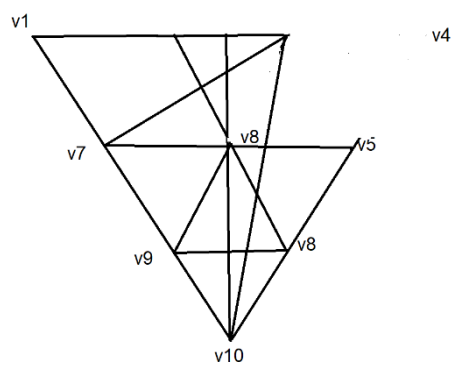
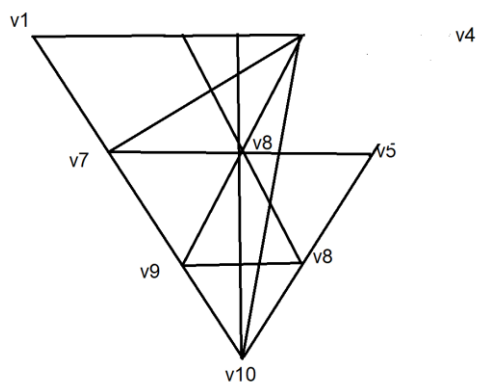
Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.

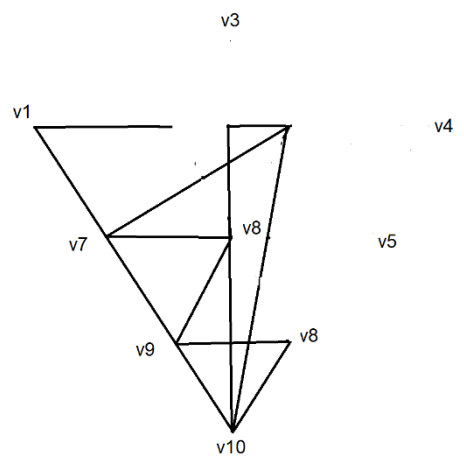
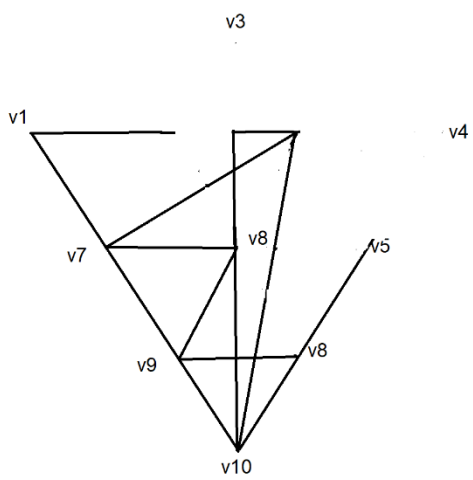
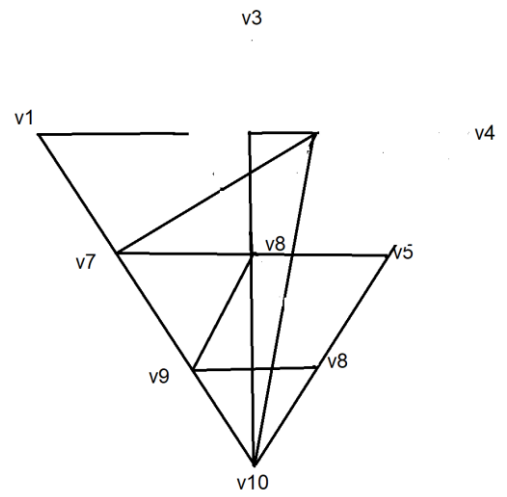
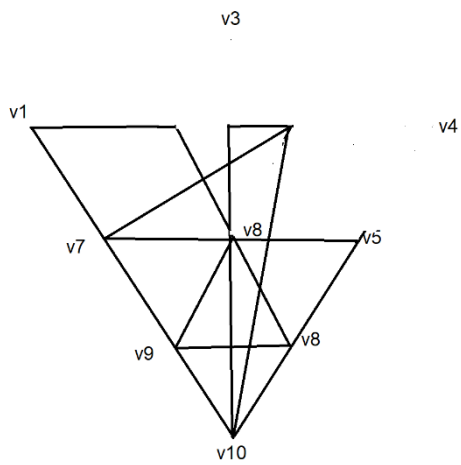


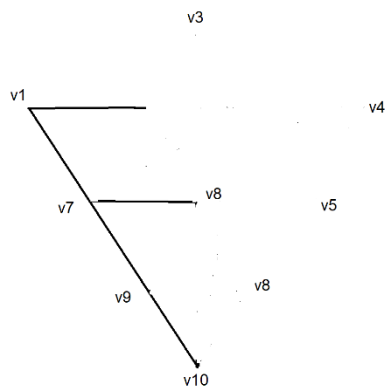
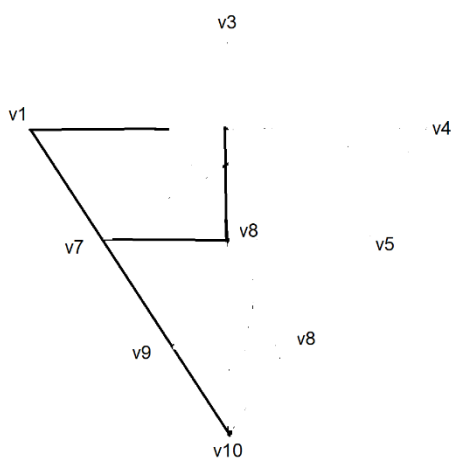
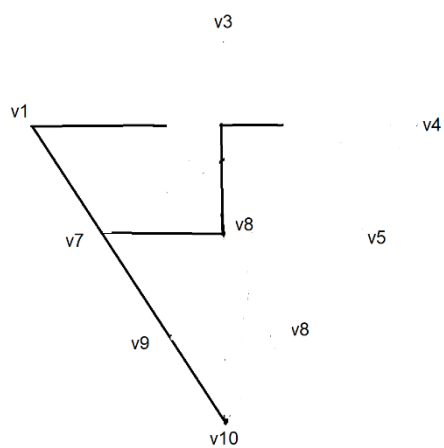
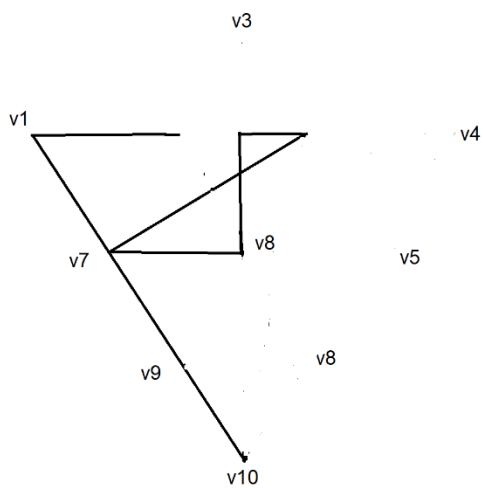
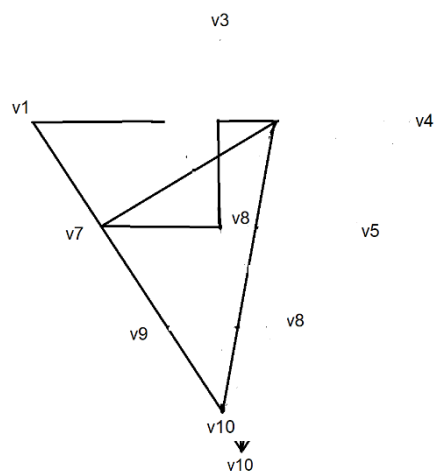
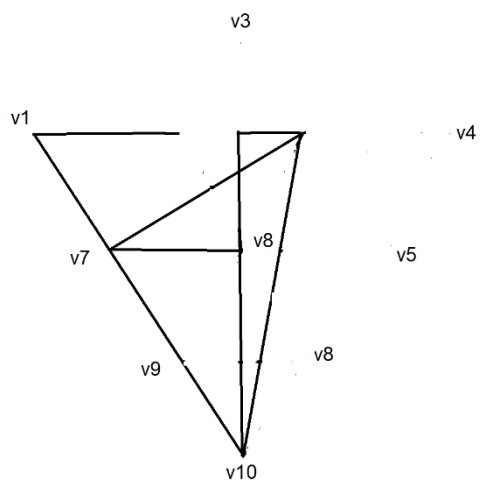


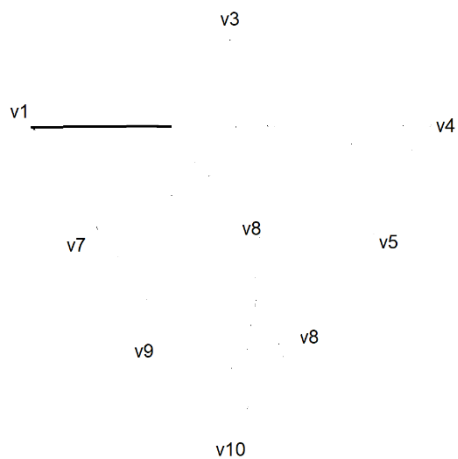
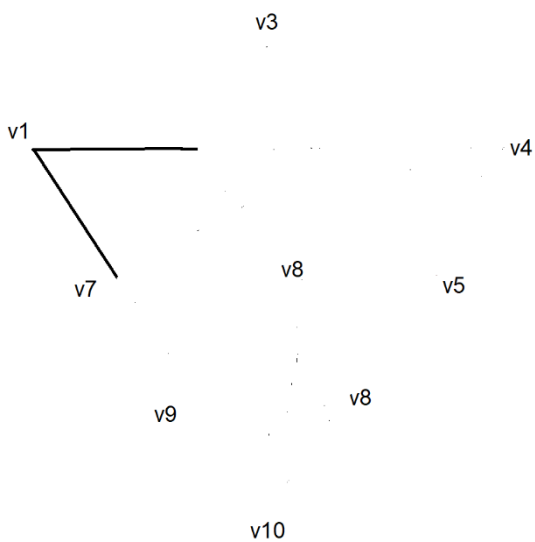
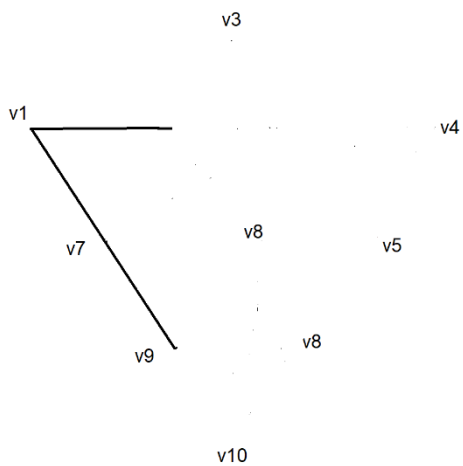
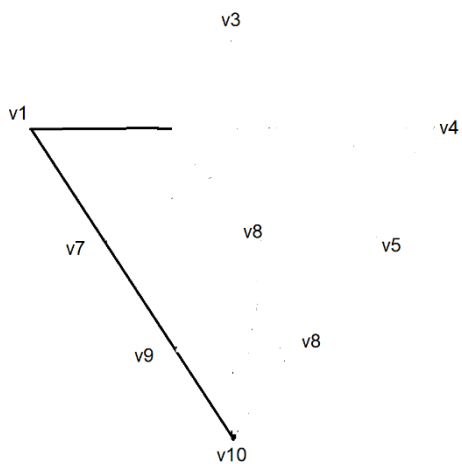
v3

v3



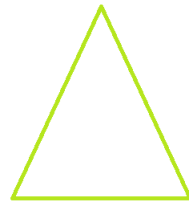
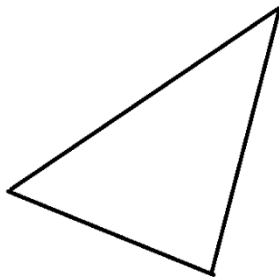
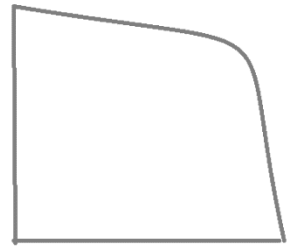
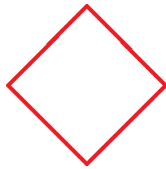
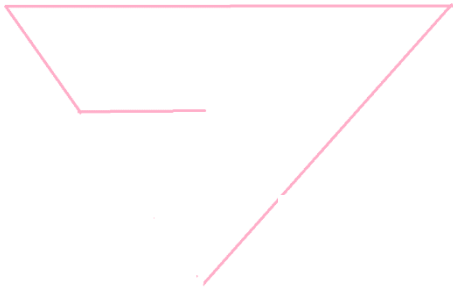








6)



Завдання №9 Спростити формули (привести їх до скороченої ДНФ).

15.  $x\bar{z} \vee xy \vee yz$

X	Y	Z	$\neg Z$	$X\neg Z$	XY	YZ	$X\neg Z \vee XY \vee YZ$
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	1	0
1	0	1	0	0	0	0	0
1	1	0	1	1	1	0	1
1	1	1	0	0	1	1	1

$$F = \neg XYZ \vee X\neg Y\neg Z \vee XY\neg Z \vee XYZ$$

$$X\neg Y\neg Z \quad X\neg Z$$

$$\neg XYZ \quad YZ$$

$$XY\neg Z \quad XY$$

$$XYZ$$

$$F = X\neg Z \vee YZ \vee XY$$



Коди

Флері

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int a[107][107],m[107],n,s,m1,s2,f2;
4  stack <int> st;
5  vector <int> res;
6
7  int main() {
8      cin>>n>>m1;
9      for(int i=1;i<=m1;i++){
10         cin>>s2>>f2;
11         a[s2][f2] = 1;
12         a[f2][s2] = 1;}
13     st.push (1);
14     while (!st.empty()) {
15         int v = st.top();
16         int i;
17         for (i=1; i<=n; ++i)
18             if (a[v][i])
19                 break;
20         if (i == n+1) {
21             res.push_back (v);
22             st.pop();
23         }
24         else {
25             --a[v][i];
26             --a[i][v];
27             st.push (i);
28         }
29     }
30     for(int i=0;i<res.size();i++){
31         cout<<res[i]<<' ';
32     }
33 }
34
```

11 22

1 2

1 7

2 3

2 6

2 11

3 11

3 6

3 5

4 11

4 5

5 6

5 8

6 11

6 7

6 10

6 8

6 9

7 11

7 9

8 10

9 10

10 11

1 8 9 7 11 10 6 9 7 6 8 5 6 11 3 6 2 11 4 5 3 2 1

Process returned 0 (0x0) execution time : 1.389 s

Press any key to continue.

Пріма

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int ways[14], a[14][14], vis[14] = {0}, tops[14] = {0}, topNumb = 0, val, s, f;
4  int min_array(int n){
5      int min = INT_MAX, nmin = 0;
6      for(int i=1; i<=n; i++){
7          if(ways[i]<min && vis[i]==0){
8              min = ways[i]; nmin = i;
9          }
10     }
11     return nmin;
12 }
13 int main() {
14     int n, m;
15
16     cout<<"Enter tops and edges of graph\n";
17     cin>>n>>m;
18     cout<<"Enter number of start top, end top and edge weight between them\n";
19     for(int i=0; i<=13; i++)
20         ways[i] = INT_MAX;
21     for(int i = 1; i<=m; i++){
22         cin>>s>>f>>val;
23         a[s][f] = val;
24         a[f][s] = val;
25     }
26     ways[1] = 0;
27     int current = 1;
28     while(current!=0){
29         vis[current] = 1;
30         for(int i=1; i<=n; i++){
31             if(a[current][i]!=0 && vis[i] ==0 ) ways[i] = min(ways[i], a[current][i]);
32         }
33         topNumb ++;
34         tops[topNumb] = current;
35         current = min_array(n);
36     }
```

```

36     }
37     cout<<"The order of tops after algorithm passing\n";
38     int sum = 0;
39     for(int i=1;i<n;i++){
40         sum += ways[i];
41         cout<<tops[i]<<" -> ";
42     }
43     cout<<tops[n];
44     cout<<"\nLength of minimal frame: "<<sum + ways[n];
45 }
46

```

```

Enter amount of tops and edges of graph
11 18
Enter number of start top,end top and edge weight between them
1 2 1
1 3 2
1 4 4
2 5 3
3 5 7
3 6 5
4 6 2
4 7 3
5 8 4
5 9 6
6 8 7
6 10 3
7 9 5
7 10 7
8 11 4
9 11 1
10 11 4
2 7 2
The order of tops after algorithm passing
1 -> 2 -> 3 -> 7 -> 4 -> 6 -> 5 -> 10 -> 8 -> 11 -> 9
Length of minimal frame: 25
Process returned 0 (0x0)   execution time : 127.738 s
Press any key to continue.

```

## Пошук вглиб

```
1  #include <vector>
2  #include <stdio.h>
3  #include <queue>
4  #include <iostream>
5  using namespace std;
6
7  const int n = 9;
8  const int m = 9;
9
10 int iArr[n][m] = {0,1,0,0,0,0,0,1,1,
11                  1,0,1,0,0,0,1,1,0,
12                  0,1,0,1,0,0,1,0,0,
13                  0,0,1,0,1,0,1,0,0,
14                  0,0,0,1,0,1,1,0,0,
15                  0,0,0,0,1,0,1,1,0,
16                  0,1,1,1,1,1,0,1,0,
17                  1,1,0,0,0,1,1,0,0,
18                  1,0,0,0,0,0,0,0,0 };
19
20 bool used[n];
21
22 int j = 0;
23 int r = 0;
24 int i = 0;
25 int k = 0;
26
27 void dfs(int t) {
28     used[t] = true;
29
30     int p;
31
32     for (i = k; i < n; i++)
33     {
```

```

35     j = r;
36     if ((iArr[i][j] != 0) && (!used[i]))
37     {
38         used[i] = true;
39         p = i;
40
41         cout << i << " ";
42
43         for (j = 0; j < m; j++)
44         {
45             i == p;
46             if (iArr[i][j] != 0)
47             {
48                 r = j;
49
50                 for (k = 0; k < n; k++)
51                 {
52                     j == r;
53
54                     if ((iArr[k][j] != 0) && (!used[k]))
55                     {
56                         dfs(i);
57                     }
58                 }
59             }
60         }
61     }
62 }
63 }
64

```

```

66 int main()
67 {
68
69     for (int i = 0; i < n; i++)
70     {
71         used[i] = false;
72         for (int j = 0; j < m; j++)
73             cout << " " << iArr[i][j];
74         cout << endl;
75     }
76
77
78     int from;
79     cout << "From >> ";
80     cin >> from;
81
82     cout << "Order: " << endl;
83
84     dfs(from);
85
86     cout << endl;
87
88     return 0;}
89

```

```

0 1 0 0 0 0 0 1 1
1 0 1 0 0 0 1 1 0
0 1 0 1 0 0 1 0 0
0 0 1 0 1 0 1 0 0
0 0 0 1 0 1 1 0 0
0 0 0 0 1 0 1 1 0
0 1 1 1 1 1 0 1 0
1 1 0 0 0 1 1 0 0
1 0 0 0 0 0 0 0 0
From >>

```

## Пошук в шир

```

1  |#include <vector>
2  |#include <stdio.h>
3  |#include <queue>
4  |#include <iostream>
5
6  |using namespace std;
7
8  |int main ()
9  |- {
10 |
11 |     vector < vector<int> > g;
12 |     const int n = 9;
13 |     int s = 0;
14 |
15 |     int Adj[n][n] = {
16 |         {0,1,0,0,0,0,0,1,1},
17 |         {1,0,1,0,0,0,1,1,0},
18 |         {0,1,0,1,0,0,1,0,0},
19 |         {0,0,1,0,1,0,1,0,0},
20 |         {0,0,0,1,0,1,1,0,0},
21 |         {0,0,0,0,1,0,1,1,0},
22 |         {0,1,1,1,1,1,0,1,0},
23 |         {1,1,0,0,0,1,1,0,0},
24 |         {1,0,0,0,0,0,0,0,0},
25 |     };
26 |     for (int i = 0; i < n; i++)
27 |     {
28 |         g.push_back(vector<int>());
29 |         for(int j = 0; j < n; j++)
30 |         {
31 |             g[i].push_back(0);
32 |             g[i][j]=Adj[i][j];
33 |         }

```

```

34     queue<int> q;
35     q.push (s);
36     vector<bool> used (n);
37     vector<int> d (n), p (n);
38     used[s] = true;
39     p[s] = -1;
40     while (!q.empty())
41     {
42         int v = q.front();
43         for (size_t i = 0; i < g[v].size(); ++i)
44         {
45             if (!used[i] && g[v][i])
46             {
47                 used[i] = true;
48                 q.push (i);
49                 d[i] = d[v] + 1;
50                 p[i] = v;
51             }
52         }
53         q.pop();
54     }
55     for (int i = 0; i < n; i++)
56         cout << d[i] << " ";
57     cout << endl;
58     for (int i = 0; i < n; i++)
59         cout << p[i] << " ";
60
61     cout << endl;
62     system("pause");
63     return 0;
64 }

```

```

0 1 2 3 3 2 2 1 1
-1 0 1 2 6 7 1 0 0
Press any key to continue . . .

```

## Краскал

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      int n,m;
6      cout<<"Enter number of tops and edges\n";
7      cin>>n>>m;
8      int cost = 0;
9      vector < pair<int,int> > res;
10     vector < pair < int, pair<int,int> > > g (m);
11     cout<<"Enter start and finish edges\n";
12     for(int i=0;i<m;i++){
13         cin>>g[i].first>>(g[i].second).first>>(g[i].second).second;
14     }
15     sort (g.begin(), g.end());
16     vector<int> tree_id (n);
17     for (int i=0; i<n; ++i)
18         tree_id[i] = i;
19     for (int i=0; i<m; ++i)
20     {
21         int a = g[i].second.first, b = g[i].second.second, l = g[i].first;
22         if (tree_id[a] != tree_id[b])
23         {
24             cost += l;
25             res.push_back (make_pair (a, b));
26             int old_id = tree_id[b], new_id = tree_id[a];
27             for (int j=0; j<n; ++j)
28                 if (tree_id[j] == old_id)
29                     tree_id[j] = new_id;
30         }
31     }
32     cout<<"Edges in minimal tree\n";
33     for(int i=0;i<res.size();i++)
34         cout<<res[i].first<<' '<<res[i].second<<endl;
35 }
```



```
Enter amount of tops and edges of graph
11 18
Enter number of start top,end top and edge weight between them
1 2 1
1 3 2
1 4 4
2 5 3
3 5 7
3 6 5
4 6 2
4 7 3
5 8 4
5 9 6
6 8 7
6 10 3
7 9 5
7 10 7
8 11 4
9 11 1
10 11 4
2 7 2
The order of tops after algorithm passing
1 -> 2 -> 3 -> 7 -> 4 -> 6 -> 5 -> 10 -> 8 -> 11 -> 9
Length of minimal frame: 25
Process returned 0 (0x0)   execution time : 127.738 s
Press any key to continue.
```

## Дейкстри

```
1  #include <iostream>
2  #include <cstdio>
3  using namespace std;
4  int n,i,j,a[101][101],b[101],c[101]={0},x,y,path[102]={0},from[102]={0};
5  void dejkstra(int v)
6  {
7      c[v]=1;
8      for(i=1;i<=n;i++)
9          if(a[v][i]!=-1 && c[i]==0 && v!=i)
10     {
11         if(b[v]+a[v][i]<b[i]){ path[i]=v;
12             b[i]=min(b[i],b[v]+a[v][i]);from[i] = v;}
13     }
14 }
15 int min_ar()
16 {
17     int min=1000000,nm=0;
18     for(i=1;i<=n;i++)
19         if(min>b[i] && c[i]==0){min=b[i];nm=i;}
20     return nm;
21 }
22
23 int out(int v){
24     cout<<endl<<"pair"<<endl;
25     cout<<v<<' '<<from[v]<<' ';
26     return from[v];
27 }
28
```

```

29     int main()
30     {
31
32         cout<<"Enter number of tops, start and finish tops\n";
33
34         cin>>n>>x>>y;
35         cout<<"Enter matrix\n";
36         for(i=1;i<=n;i++)
37         {
38             for(j=1;j<=n;j++){
39                 cin>>a[i][j];
40             }
41
42             b[i]=10000000;
43         }
44         if(a[x][y]==0){cout<<0;return 0;}
45
46         b[x]=0;
47         path[1]=x;
48         int t=1;
49         while(t==1)
50         {
51             t=0;
52             int v=min_ar();
53             if(c[v]==0 && v>0 ) {dejkstra(v); t=1;}
54         }
55
56         cout<<"The shortest way: "<<b[y]<<endl;
57         cout<<"Tops on the shortest way: ";
58         int curr = y;
59         while(curr!=x){
60             curr = out(curr);
61         }
62

```