МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Розрахункова робота

з дисципліни «Дискретна математика»

Виконав:

Студент групи КН-112

Шкляров Віталій

Викладач:

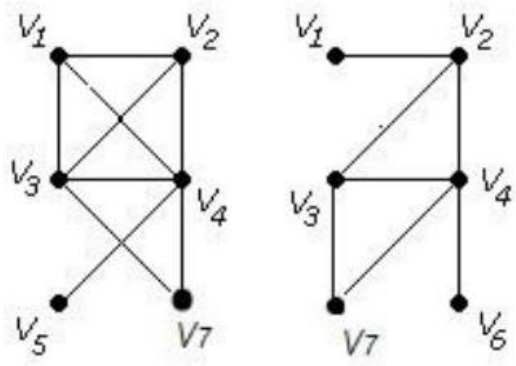
Мельникова Н. І.

Львів 2019

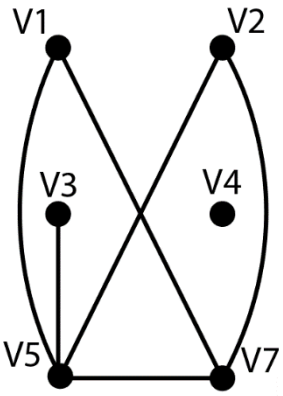
Варіант 15

Завдання № 1

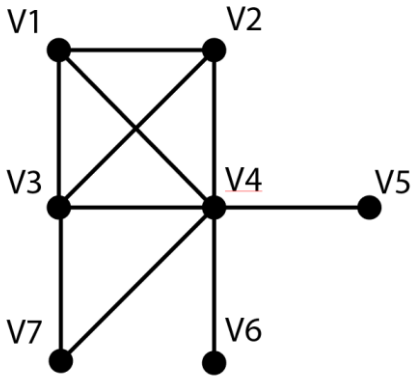
Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об’єднання графів, 3) кільцеву сумму G 1 та G 2 (G1 + G2), 4) розмножити вершину у другому графі, 5) виділити підграф А - що скадається з 3-х вершин в G1 6) добуток графів.



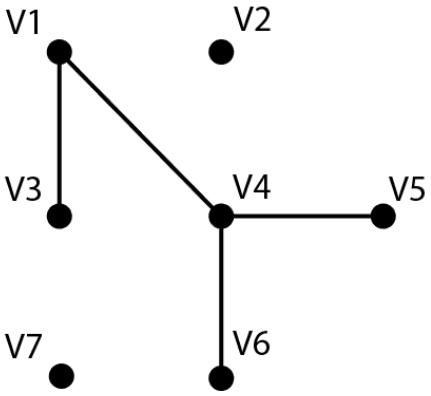
1)



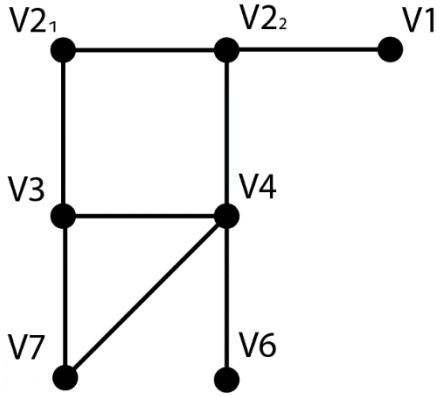
2)



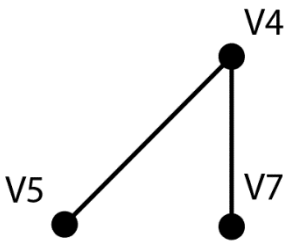
3)



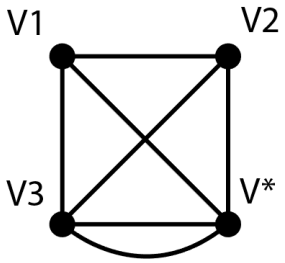
4)



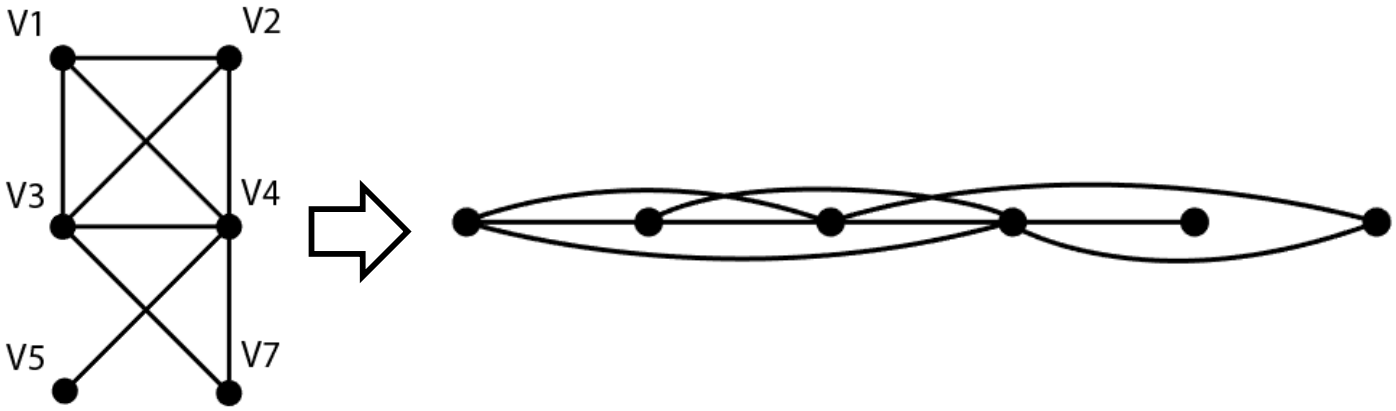
5) Підграф А

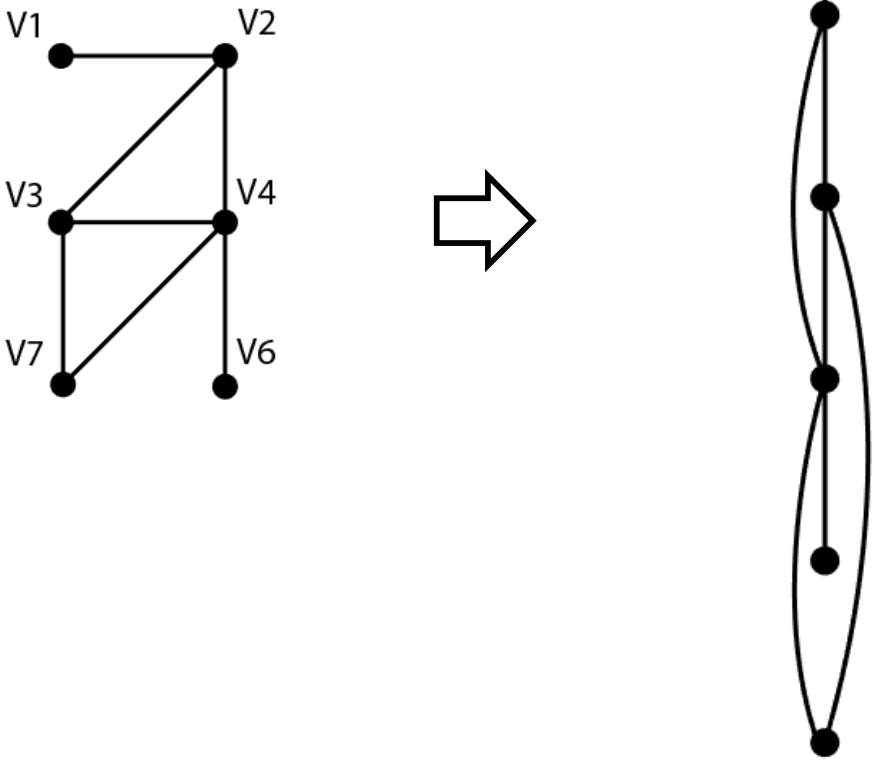


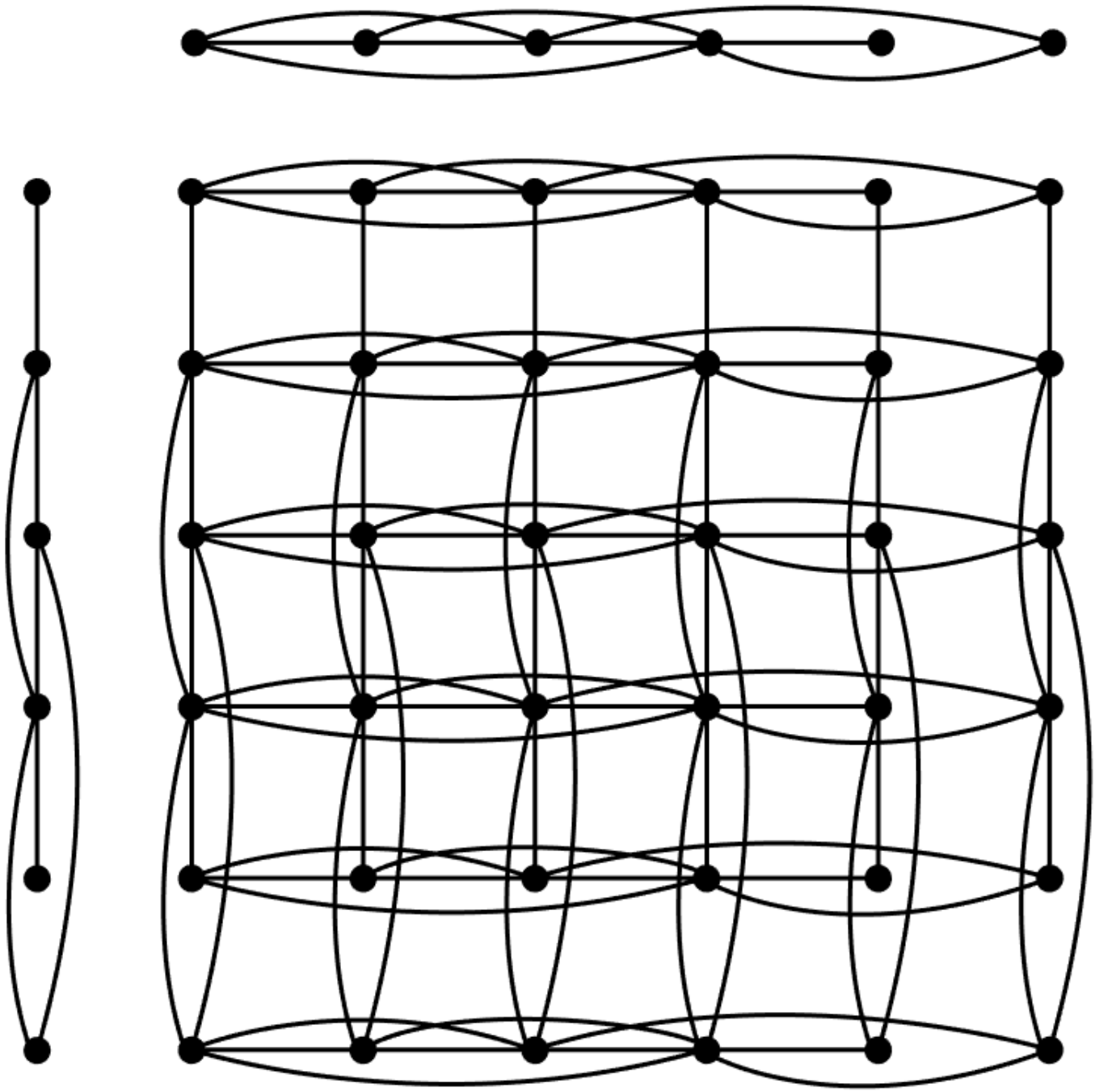
Новий граф



6)

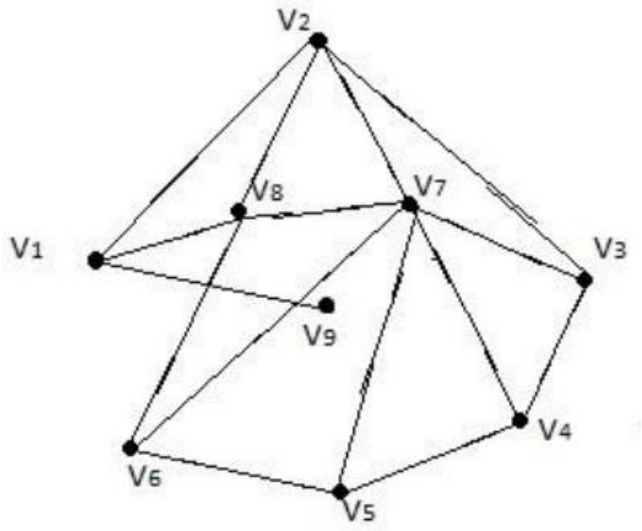






Завдання № 2

Скласти таблицю суміжності для орграфа.



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

Завдання № 3

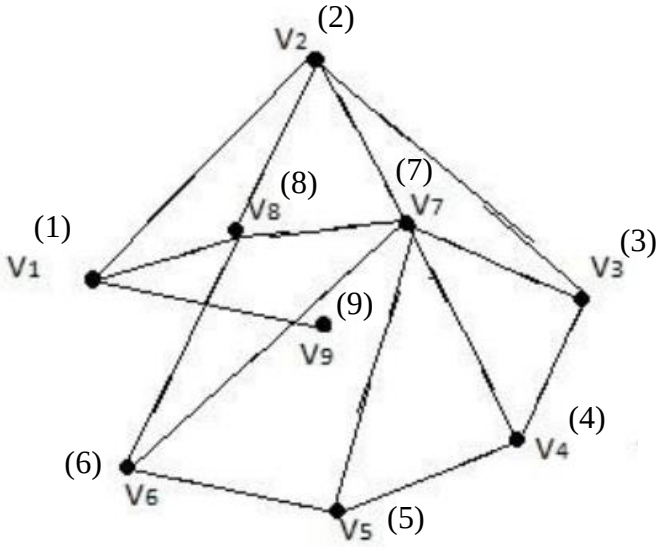
Для графа з другого завдання знайти діаметр.

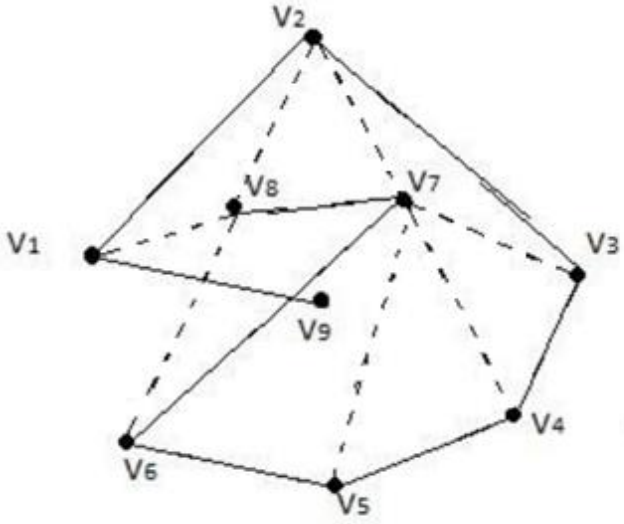
	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	2	3	3	2	2	1	1
V2	1	0	1	2	2	2	1	1	2
V3	2	1	0	1	2	2	1	2	3
V4	3	2	1	0	1	2	1	2	4
V5	3	2	2	1	0	1	1	2	4
V6	2	2	2	2	1	0	1	1	3
V7	2	1	1	1	1	1	0	1	3
V8	1	1	2	2	2	1	1	0	2
V9	1	2	3	4	4	3	3	2	0

Діаметр = 4

Завдання № 4

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





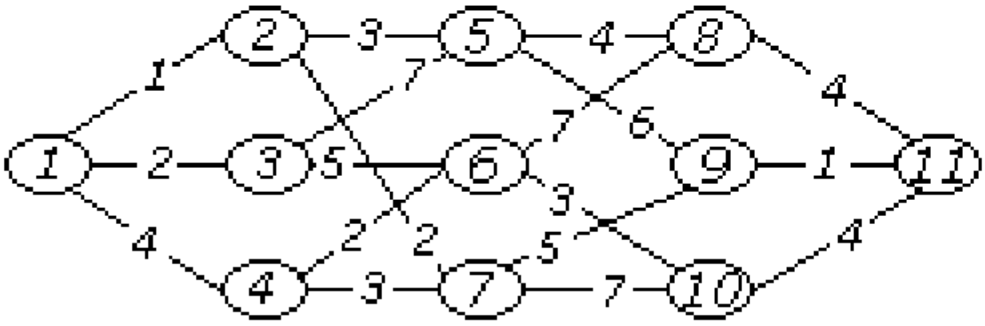
Вершина	DFS-номер	Вміст стеку
V1	1	V1
V2	2	V1 V2
V3	3	V1 V2 V3
V4	4	V1 V2 V3 V4
V5	5	V1 V2 V3 V4 V5
V6	6	V1 V2 V3 V4 V5 V6
V7	7	V1 V2 V3 V4 V5 V6 V7
V8	8	V1 V2 V3 V4 V5 V6 V7 V8
-	-	V1 V2 V3 V4 V5 V6 V7
-	-	V1 V2 V3 V4 V5 V6
-	-	V1 V2 V3 V4 V5
-	-	V1 V2 V3 V4
-	-	V1 V2 V3
-	-	V1 V2
-	-	V1
V9	9	V1 V9
-	-	V1
-	-	Ø

```cpp
#include <iostream>

using namespace std;
const int n = 9;
bool *visited = new bool[n];

int graph[n][n] =
        {
                {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}
        };

void DFS(int st) {
    cout << st + 1 << " ";
    visited[st] = true;
    for (int r = 0; r <= n; r++)
        if ((graph[st][r] != 0) && (!visited[r])) {
            DFS(r);
        }
}

int main() {
    int start;
    cout << "Matrix: " << endl;
    for (int i = 0; i < n; i++) {
        visited[i] = false;
        for (int j = 0; j < n; j++)
            cout << " " << graph[i][j];
        cout << endl;
    }
    cout << "First Vertex:";
    cin >> start;

    bool *vis = new bool[n];
    cout << "Result: ";
    DFS(start - 1);
    delete[]visited;
}
```

```
Matrix:
 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
First Vertex:1
Result: 1 2 3 4 5 6 7 8 9
Process finished with exit code 0
```

Завдання № 5

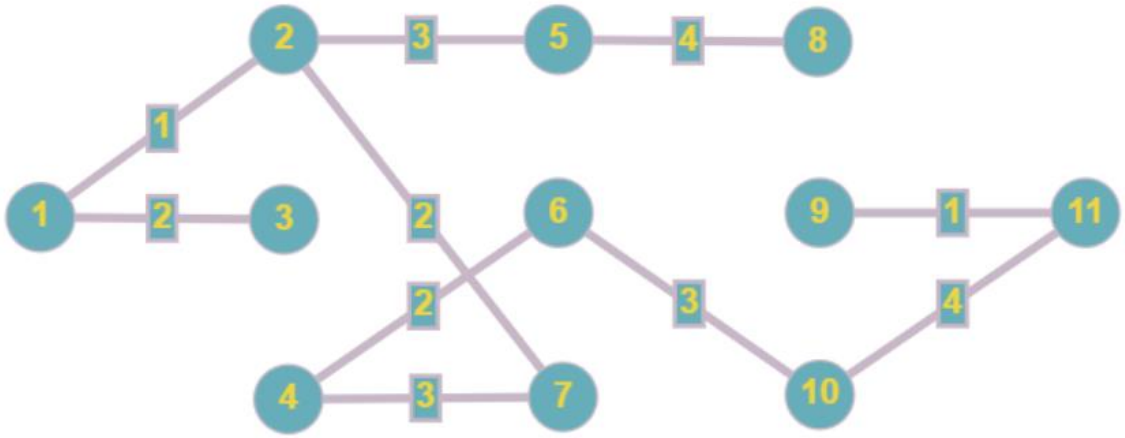
Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.



Метод Краскала

V : {1, 2, 9, 11, 3, 4, 6, 7, 5, 10, 8}

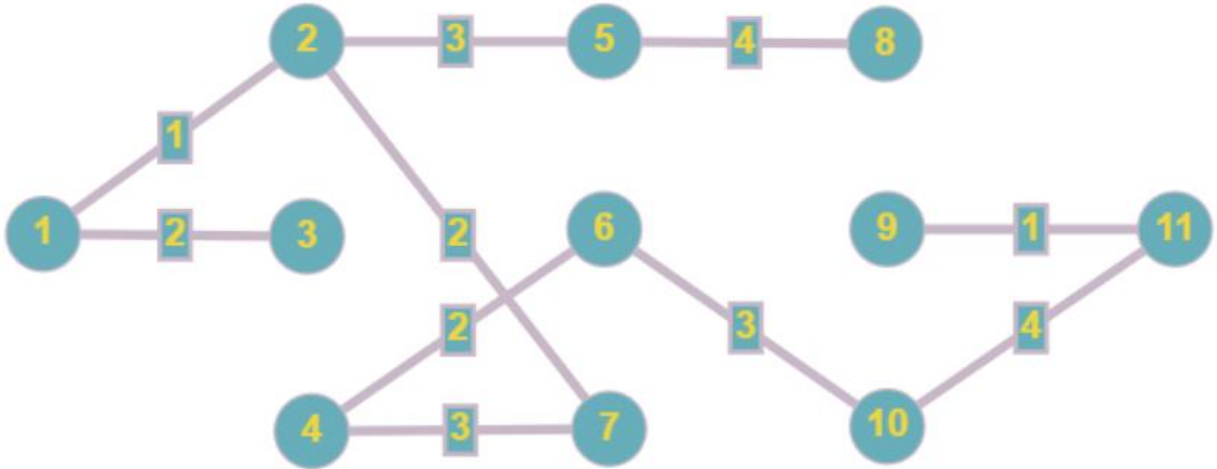
E : {(1,2); (9,11); (1,3); (4,6); (2,7); (4,7); (2,5); (6,10); (5,8); (10,11)}



Метод Прима

V : {1, 2, 3, 7, 5, 4, 6, 10, 8, 11, 9}

E : {(1,2); (1,3); (2,7); (2,5); (7,4); (4,6); (6,10); (5,8); (10,11); (11,9)}



Метод Прима

```cpp
#include <iostream>

using namespace std;

int main() {
    int v, count = 0, min = 0, k, t;
    bool check = false;
    cout <<"Enter quantity of vertex : ";
    cin >> v;
    int *tops = new int[v];
    int **g = new int *[v];
    int **r = new int *[v - 1];

    for (int j = 0; j < v; j++) {
        g[j] = new int[v];
    }
    for (int j = 0; j < v - 1; j++) {
        r[j] = new int[2];
    }
    for (int a = 0; a < v; a++) {
        for (int j = 0; j < v; j++) {
            cin >> g[a][j];
        }
    }
    tops[count] = 1;
    count++;
    for (int i = 0; count < v; i++) {
        for (int j = 0; j < count; j++) {
            for (int a = 0; a < v; a++) {
                for (int m = 0; m < count; m++) {
                    if (tops[m] == a + 1) {
                        check = true;
                    }
                }
                if (check) {
                    check = false;
                    continue;
                }
                if (min == 0 && g[tops[j] - 1][a] > 0) {
                    min = g[tops[j] - 1][a];
                    k = r[count - 1][0] = tops[j];
                    t = r[count - 1][1] = a + 1;
                    continue;
                }
                if (g[tops[j] - 1][a] > 0 && g[tops[j] - 1][a] < min) {
                    min = g[tops[j] - 1][a];
                    k = r[count - 1][0] = tops[j];
                    t = r[count - 1][1] = a + 1;
                }
            }
        }
        g[k - 1][t - 1] = 0;
        g[t - 1][k - 1] = 0;

        tops[count] = t;
        count++;
        min = 0;
    }
    cout << "V: { ";
    for (int j = 0; j < v; j++) {
        cout << tops[j] << ", ";
    }
    cout << "}";
    cout << endl << "E:{ ";
    for (int j = 0; j < v - 1; j++) {
        cout << "( " << r[j][0] << ", " << r[j][1] << " ), ";
    }
    cout << "}";


    delete [] tops;
    for (int i = 0; i < v; i++)
        delete [] g[i];
    for (int i = 0; i < v; i++)
        delete [] r[i];

    return 0;
}
```

Вивід:

```
Enter quantity of vertex :11
0 1 2 4 0 0 0 0 0 0 0
1 0 0 0 3 0 2 0 0 0 0
2 0 0 0 7 5 0 0 0 0 0
4 0 0 0 0 2 3 0 0 0 0
0 3 7 0 0 0 0 4 6 0 0
0 0 5 2 0 0 0 7 0 3 0
0 2 0 3 0 0 0 0 5 7 0
0 0 0 0 4 7 0 0 0 0 4
0 0 0 0 6 0 5 0 0 0 1
0 0 0 0 0 3 7 0 0 0 4
0 0 0 0 0 0 0 4 1 4 0
 V: { 1, 2, 3, 7, 5, 4, 6, 10, 8, 11, 9, }
E:{ ( 1, 2 ), ( 1, 3 ), ( 2, 7 ), ( 2, 5 ), ( 7, 4 ),
( 4, 6 ), ( 6, 10 ), ( 5, 8 ), ( 10, 11 ), ( 11, 9 ), }
```

Метод Краскала

```cpp
#include <iostream>

using namespace std;

int create(int n, int A[11][11]) {
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 11; j++) {
            A[i][j] = 0;
        }
    }
    for (int i = 0; i < 11; i++) {
        A[i][i] = i + 1;
    }
    return A[11][11];
}

void RemoveDuplicats(int n, int A[11][11]) {
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 11; j++) {
            if (j < i) {
                A[i][j] = 0;
            }
        }
    }
}

int NotOne(int n, int A[11][11], int f, int s) {
    int tmp, tmp1;
    for (int i = 0; i < 11; i++) {
        tmp = tmp1 = 0;
        for (int j = 0; j < 11; j++) {
            if (A[i][j] == f) {
                tmp = 1;
            }
        }
        for (int k = 0; k < 11; k++) {
            if (A[i][k] == s) {
                tmp1 = 1;
            }
        }
        if (tmp && tmp1) {
            return 0;
        }
    }


    return 1;
}
```

```cpp
void add(int n, int A[11][11], int f, int s) {
    int tmp;
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 11; j++) {
            if (A[i][j] == s) {
                tmp = i;
            }
        }
    }
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 11; j++) {
            if (A[i][j] == f) {
                for (int k = 0; k < 11; k++) {
                    if (A[tmp][k]) {
                        A[i][k] = A[tmp][k];
                        A[tmp][k] = 0;
                    }
                }
            }
        }
    }

}

int main() {
    int MS[11][11];
    int ms;
    for (int l = 0; l < 11; ++l) {
        for (int i = 0; i < 11; ++i) {
            cin >> MS[l][i];
        }
    }
    RemoveDuplicats(11, MS);
    int B[11][11];
    create(11, B);
    cout << endl << "New Tree:" << endl;
    for (int i = 1; i <= 7; i++) {
        for (int j = 1; j <= 11; j++) {
            for (int k = 1; k <= 11; k++) {
                if (MS[j - 1][k - 1] == i && NotOne(11, B, j, k)) {
                    add(11, B, j, k);
                    cout << " (" << j << ";" << k << ") ";
                }

            }
        }
    }


    return 0;
}
```

```
0 1 2 4 0 0 0 0 0 0 0
1 0 0 0 3 0 2 0 0 0 0
2 0 0 0 7 5 0 0 0 0 0
4 0 0 0 0 2 3 0 0 0 0
0 3 7 0 0 0 0 4 6 0 0
0 0 5 2 0 0 0 7 0 3 0
0 2 0 3 0 0 0 0 5 7 0
0 0 0 0 4 7 0 0 0 0 4
0 0 0 0 6 0 5 0 0 0 1
0 0 0 0 0 3 7 0 0 0 4
0 0 0 0 0 0 0 4 1 4 0

New Tree:
 (1;2)  (9;11)  (1;3)  (2;7)  (4;6)  (2;5)  (4;7)  (6;10)  (5;8)  (8;11)
```

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

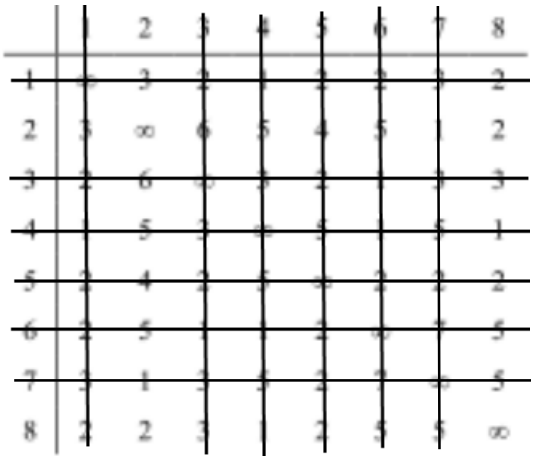
	1	2	3	4	5	6	7	8
1	∞	3	2	1	2	2	3	2
2	3	∞	6	5	4	5	1	2
3	2	6	∞	3	2	1	3	3
4	1	5	3	∞	5	1	5	1
5	2	4	2	5	∞	2	2	2
6	2	5	1	1	2	∞	7	5
7	3	1	3	5	2	7	∞	5
8	2	2	3	1	2	5	5	∞

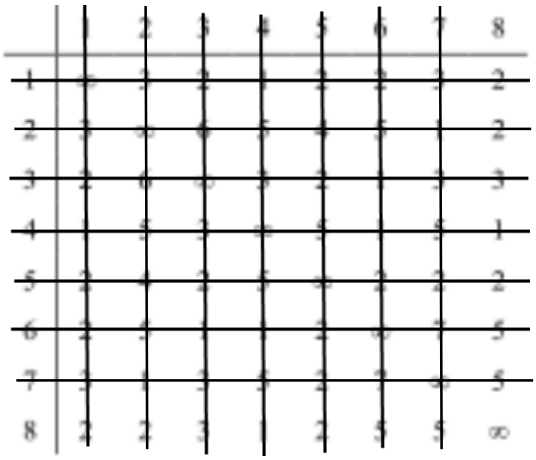
	1	2	3	4	5	6	7	8
1	∞	3	2	1	2	2	3	2
2	3	∞	6	5	4	5	1	2
3	2	6	∞	3	2	1	3	3
4	1	5	3	∞	5	1	5	1
5	2	4	2	5	∞	2	2	2
6	2	5	1	1	2	∞	7	5
7	3	1	3	5	2	7	∞	5
8	2	2	3	1	2	5	5	∞

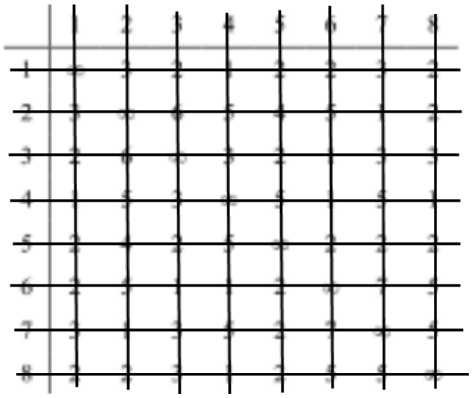
	1	2	3	4	5	6	7	8
1	∞	3	2	1	2	2	3	2
2	3	∞	6	5	4	5	1	2
3	2	6	∞	3	2	1	3	3
4	1	5	3	∞	5	1	5	1
5	2	4	2	5	∞	2	2	2
6	2	5	1	1	2	∞	7	5
7	3	1	3	5	2	7	∞	5
8	2	2	3	1	2	5	5	∞

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

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







Порядок: 1 4 6 3 5 7 2 8 1

1+1+1+2+2+1+2+2=12

1 4 6 3 5 7 2 8 1 {12}

1 4 8 2 7 5 3 6 1 {12}

1 4 8 2 7 5 6 3 1 {12}

2 7 5 3 6 4 1 8 2 {12}

2 7 5 6 3 1 4 8 2 {12}

3 6 4 1 8 2 7 5 3 {12}

3 6 4 8 2 7 5 1 3 {12}

4 1 3 6 5 7 2 8 4 {12}

4 1 6 3 5 7 2 8 4 {12}

4 1 8 2 7 5 3 6 4 {12}

4 6 3 1 5 7 2 8 4 {12}

4 6 3 5 7 2 8 1 4 {12}

4 8 2 7 5 1 3 6 4 {12}

4 8 2 7 5 3 6 1 4 {12}

4 8 2 7 5 6 3 1 4 {12}

5 3 6 4 1 8 2 7 5 {12}

5 6 3 1 4 8 2 7 5 {12}

5 7 2 8 4 1 3 6 5 {12}

5 7 2 8 4 1 6 3 5 {12}

5 7 2 8 4 6 3 1 5 {12}

6 3 1 4 8 2 7 5 6 {12}

6 3 5 7 2 8 4 1 6 {12}

6 4 1 8 2 7 5 3 6 {12}

6 4 8 2 7 5 1 3 6 {12}

7 2 8 4 1 3 6 5 7 {12}

7 2 8 4 1 6 3 5 7 {12}

7 2 8 4 6 3 1 5 7 {12}

8 4 1 3 6 5 7 2 8 {12}

8 4 1 6 3 5 7 2 8 {12}

8 4 6 3 1 5 7 2 8 {12}

```cpp
#include<iostream>
#include<vector>

using namespace std;
int counter = 0, Inf = 9999;

bool check(vector<int> q, int Node) {
    for (auto i = q.begin(); i != q.end(); i++)
        if (*i == Node)return false;
    return true;
}


int F_Min(vector<int> *q, int **arr, int n, int i) {
    int min = 999;
    for (int j = 0; j < n; j++) {
        if (arr[i][j] < min && arr[i][j] != 0 && check((*q), j))min = arr[i][j];
    }
    return min;
}
```

```cpp
void Find(vector<int> *q, int **arr, int n, int pos, vector<int> *qq) {
    int min;
    for (int i = pos, k = 0; k < 1; i++, k++) {
        min = F_Min(q, arr, n, i);
        for (int j = 0; j < n; j++) {
            if (arr[i][j] == min && check((*q), j)) {
                (*q).push_back(j);
                Find(q, arr, n, j, qq);
            }
        }
        if (q->size() == n) {
            (*q).push_back((*q)[0]);
            counter = 0;

            for (int l = 1; l <= n; l++) {
                counter += arr[(*q)[l - 1]][(*q)[l]];
            }

            if (Inf == counter) {
                for (int b = 0; b <= n; b++) {
                    (*qq).push_back((*q)[b]);
                }
                (*qq).push_back(counter);
            } else if (Inf > counter) {
                (*qq).clear();

                for (int b = 0; b <= n; b++) {
                    (*qq).push_back((*q)[b]);

                }
                (*qq).push_back(counter);

                Inf = counter;
            }
            q->pop_back();
        }
    }
    q->pop_back();
}
```

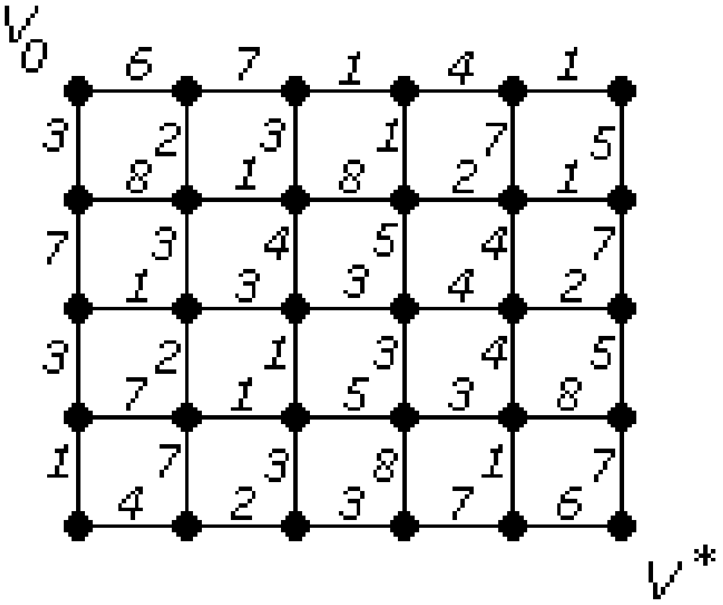
```cpp
int main() {
    int n;
    cin >> n;
    int **arr = new int *[n];
    for (int i = 0; i < n; i++) {
        arr[i] = new int[n];
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> arr[i][j];
        }
    }

    vector<int> q;
    vector<int> qq;
    cout << endl;

    for (int i = 0; i < n; i++) {
        q.clear();
        q.push_back(i);
        Find(&q, arr, n, i, &qq);
    }

    for (int i = 1; i <= qq.size(); i++) {
        if (i != 0 && i % (n + 2) == 0)
            cout << " {" << qq[i - 1] << "}" << endl;
        else
            cout << qq[i - 1] + 1 << " ";
    }
    return 0;
}
```

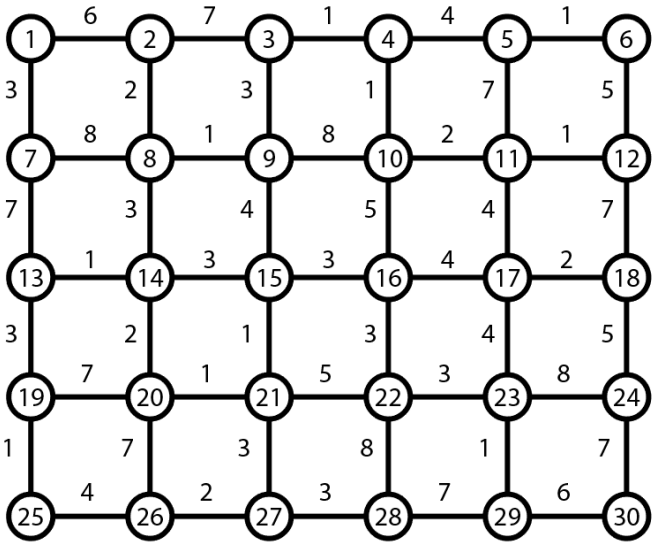
Вивід

```
1 4 6 3 5 7 2 8 1  {12}
1 4 8 2 7 5 3 6 1  {12}
1 4 8 2 7 5 6 3 1  {12}
2 7 5 3 6 4 1 8 2  {12}
2 7 5 6 3 1 4 8 2  {12}
3 6 4 1 8 2 7 5 3  {12}
3 6 4 8 2 7 5 1 3  {12}
4 1 3 6 5 7 2 8 4  {12}
4 1 6 3 5 7 2 8 4  {12}
4 1 8 2 7 5 3 6 4  {12}
4 6 3 1 5 7 2 8 4  {12}
4 6 3 5 7 2 8 1 4  {12}
4 8 2 7 5 1 3 6 4  {12}
4 8 2 7 5 3 6 1 4  {12}
4 8 2 7 5 6 3 1 4  {12}
5 3 6 4 1 8 2 7 5  {12}
5 6 3 1 4 8 2 7 5  {12}
5 7 2 8 4 1 3 6 5  {12}
5 7 2 8 4 1 6 3 5  {12}
5 7 2 8 4 6 3 1 5  {12}
6 3 1 4 8 2 7 5 6  {12}
6 3 5 7 2 8 4 1 6  {12}
6 4 1 8 2 7 5 3 6  {12}
6 4 8 2 7 5 1 3 6  {12}
7 2 8 4 1 3 6 5 7  {12}
7 2 8 4 1 6 3 5 7  {12}
7 2 8 4 6 3 1 5 7  {12}
8 4 1 3 6 5 7 2 8  {12}
8 4 1 6 3 5 7 2 8  {12}
8 4 6 3 1 5 7 2 8  {12}
```

Завдання № 7

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





Ітерація	Мітки	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	L	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q																														
2	L	0	6	∞	∞	∞	∞	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1					1																							
3	L	0	6	∞	∞	∞	∞	3	11	∞	∞	∞	∞	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1					1	7					7																	
4	L	0	6	13	∞	∞	∞	3	8	∞	∞	∞	∞	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	2				1	2					7																	
5	L	0	6	13	∞	∞	∞	3	8	9	∞	∞	∞	10	11	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	2				1	2	8				7	8																
6	L	0	6	12	∞	∞	∞	3	8	9	17	∞	∞	10	11	13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	9				1	2	8	9			7	8	9															
7	L	0	6	12	∞	∞	∞	3	8	9	17	∞	∞	10	11	13	∞	∞	∞	13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	9				1	2	8	9			7	13	9				13											
8	L	0	6	12	∞	∞	∞	3	8	9	17	∞	∞	10	11	13	∞	∞	∞	13	13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	9				1	2	8	9			7	13	9				13	14										
9	L	0	6	12	13	∞	∞	3	8	9	17	∞	∞	10	11	13	∞	∞	∞	13	13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	9	3			1	2	8	9			7	13	9				13	14										
10	L	0	6	12	13	17	∞	3	8	9	14	∞	∞	10	11	13	∞	∞	∞	13	13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	9	3	4		1	2	8	4			7	13	9				13	14										
11	L	0	6	12	13	17	∞	3	8	9	14	∞	∞	10	11	13	16	∞	∞	13	13	14	∞	∞	∞	∞	∞	∞	∞	∞	∞
	Q		1	9	3	4		1	2	8	4			7	13	9	15			13	14	15									
12	L	0	6	12	13	17	∞	3	8	9	14	∞	∞	10	11	13	16	∞	∞	13	13	14	∞	∞	∞	14	∞	∞	∞	∞	∞
	Q		1	9	3	4		1	2	8	4			7	13	9	15			13	14	15				19					
13	L	0	6	12	13	17	∞	3	8	9	14	∞	∞	10	11	13	16	∞	∞	13	13	14	∞	∞	∞	14	20	∞	∞	∞	∞
	Q		1	9	3	4		1	2	8	4			7	13	9	15			13	14	20				19	20				
14	L	0	6	12	13	17	∞	3	8	9	14	16	∞	10	11	13	16	∞	∞	13	13	14	∞	∞	∞	14	20	∞	∞	∞	∞
	Q		1	9	3	4		1	2	8	4	10		7	13	9	15			13	14	20				19	20				
15	L	0	6	12	13	17	∞	3	8	9	14	16	∞	10	11	13	16	∞	∞	13	13	14	19	∞	∞	14	20	17	∞	∞	∞
	Q		1	9	3	4		1	2	8	4	10		7	13	9	15			13	14	20	21			19	20	21			

Ітерація	Мітки	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
16	L	0	6	12	13	17	∞	3	8	9	14	16	∞	10	11	13	16	∞	∞	13	13	14	19	∞	∞	14	18	17	∞	∞	∞
	Q		1	9	3	4		1	2	8	4	10		7	13	9	15			13	14	20	21			19	25	21			
17	L	0	6	12	13	17	∞	3	8	9	14	16	17	10	11	13	16	20	∞	13	13	14	19	∞	∞	14	18	17	∞	∞	∞
	Q		1	9	3	4		1	2	8	4	10	11	7	13	9	15	11		13	14	20	21			19	25	21			
18	L	0	6	12	13	17	∞	3	8	9	14	16	17	10	11	13	16	20	∞	13	13	14	19	∞	∞	14	18	17	∞	∞	∞
	Q		1	9	3	4		1	2	8	4	10	11	7	13	9	15	16		13	14	20	16			19	25	21			
19	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	∞	13	13	14	19	∞	∞	14	18	17	∞	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16		13	14	20	16			19	25	21			
20	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	24	13	13	14	19	∞	∞	14	18	17	∞	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	12	13	14	20	16			19	25	21			
21	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	24	13	13	14	19	∞	∞	14	18	17	20	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	12	13	14	20	16			19	25	21	27		
22	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	24	13	13	14	19	∞	∞	14	18	17	20	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	12	13	14	20	16			19	25	21	27		
23	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	24	13	13	14	19	∞	∞	14	18	17	20	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	12	13	14	20	16			19	25	21	27		
24	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	24	13	13	14	19	22	∞	14	18	17	20	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	12	13	14	20	16	22		19	25	21	27		
25	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	22	13	13	14	19	22	∞	14	18	17	20	∞	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	17	13	14	20	16	22		19	25	21	27		
26	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	22	13	13	14	19	22	∞	14	18	17	20	27	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	17	13	14	20	16	22		19	25	21	27	28	
27	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	22	13	13	14	19	22	27	14	18	17	20	27	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	17	13	14	20	16	22	18	19	25	21	27	28	
28	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	22	13	13	14	19	22	27	14	18	17	20	23	∞
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	17	13	14	20	16	22	18	19	25	21	27	23	
29	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	22	13	13	14	19	22	27	14	18	17	20	23	29
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	17	13	14	20	16	22	18	19	25	21	27	23	29
30	L	0	6	12	13	17	18	3	8	9	14	16	17	10	11	13	16	20	22	13	13	14	19	22	27	14	18	17	20	23	29
	Q		1	9	3	4	5	1	2	8	4	10	11	7	13	9	15	16	17	13	14	20	16	22	18	19	25	21	27	23	29

Порядок проходження вершин

1 –> 2 –> 8 –> 9 –> 15 –> 16 –> 22 –> 23 –> 29 –> 30

```cpp
#include <iostream>

using namespace std;

int main() {
    int SIZE = 30;
    int a[SIZE][SIZE]; // матриця
    int d[SIZE]; // мінімальна відстань
    int v[SIZE]; // відвідані вершини
    int temp, minindex, min;
    int begin_index;
    cin >> begin_index;
    --begin_index;
    int end;
    cin >> end;
    --end;

    for (int i = 0; i < SIZE; ++i) {
        for (int j = 0; j < SIZE; ++j) {
            cin >> a[i][j];
        }
    }
```

```cpp
    for (int i = 0; i < SIZE; i++) {
        d[i] = 10000;
        v[i] = 1;
    }
    d[begin_index] = 0;

    do {
        minindex = 10000;
        min = 10000;
        for (int i = 0; i < SIZE; i++) {
            if ((v[i] == 1) && (d[i] < min)) {
                min = d[i];
                minindex = i;
            }
        }
        if (minindex != 10000) {
            for (int i = 0; i < SIZE; i++) {
                if (a[minindex][i] > 0) {
                    temp = min + a[minindex][i];
                    if (temp < d[i]) {
                        d[i] = temp;
                    }
                }
            }
            v[minindex] = 0;
        }
    } while (minindex < 10000);
```

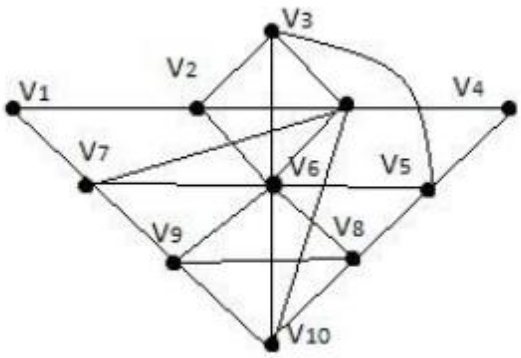
```
    int ver[SIZE];
    ver[0] = end + 1;
    int k = 1;
    int weight = d[end];

    while (end != begin_index) {
        for (int i = 0; i < SIZE; i++) {
            if (a[end][i] != 0) {
                int temp = weight - a[end][i];
                if (temp == d[i]) {
                    weight = temp;
                    end = i;
                    ver[k] = i + 1;
                    k++;
                }
            }
        }
    }

    for (int i = k - 1; i >= 0; i--) {
        cout << ver[i];
        if (i != 0)
            cout << " -> ";
    }

    return 0;
}
```

```
1
30
0 6 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 7 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 7 0 1 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 4 0 1 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 8 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 2 0 0 0 0 8 0 1 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 3 0 0 0 0 1 0 8 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 8 0 2 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 7 0 0 0 0 2 0 1 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 5 0 0 0 0 1 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 7 0 0 0 0 0 0 1 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 0 0 0 0 1 0 3 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 4 0 0 0 0 3 0 3 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 5 0 0 0 0 3 0 4 0 0 0 0 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 4 0 2 0 0 0 0 4 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 2 0 0 0 0 0 0 5 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 7 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 7 0 1 0 0 0 0 7 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 5 0 0 0 0 3 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 5 0 3 0 0 0 0 8 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 3 0 8 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 8 0 0 0 0 0 0 7
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 4 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 4 0 2 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 2 0 3 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 3 0 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 7 0 6
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 6 0
```

```
1 -> 2 -> 8 -> 9 -> 15 -> 16 -> 22 -> 23 -> 29 -> 30
Process finished with exit code 0
```

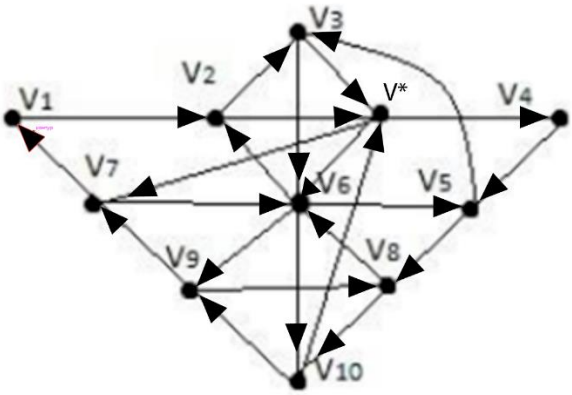
Завдання № 8

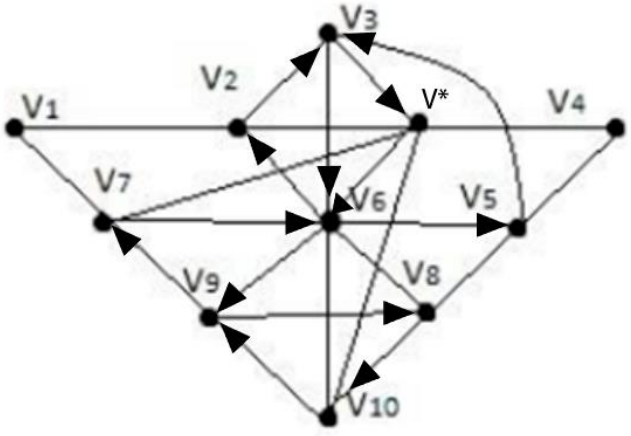
Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.



а) (V1,V2); (V2,V3); (V3,V*); (V*,V6); (V6,V2); (V2,V*); (V*,V4); (V4,V5); (V5,V8); (V8,V10); (V10,V9); (V9,V8); (V8,V6); (V6,V9); (V9,V7); (V7,V6); (V6,V5); (V5,V3); (V3,V6); (V6,V10); (V10,V*); (V*,V7); (V7,V1).

V1 –> V2 –> V3 –> V* –> V6 –> V2 –> V* –> V4 –> V5 –> V8 –> V10 –> –> V9 –> V8 –> V6 –> V9 –> V7 –> V6 –> V5 –> V3 –> V6 –> V10 –> –> V* –> V7 –> V1.





б) Виділимо елементарні цикли

а) V2 –> V3 –> V* –> V6 –> V2

б) V8 –> V10 –> –> V9 –> V8

в) V6 –> V5 –> V3 –> V6

г) V6 –> V9 –> V7 –> V6

З’єднаємо їх

V1 –> а –> V* –> V4 –> V5 –> б –> (V6 –> V9 –> V7 –> (V6) –> V5 –> –> V3 –> V6) –> V10 –> V* –> V7 –> V1

в г

```cpp
#include <iostream>
#include <vector>

using namespace std;

int main() {
    int v = 0;
    cout << "Quantity :";
    cin >> v;
    cout << "Matrix:" << endl;
    int **graph = new int *[v];
    for (int j = 0; j < v; j++) {
        graph[j] = new int[v];
    }
    for (int a = 0; a < v; a++) {
        for (int j = 0; j < v; j++) {
            cin >> graph[a][j];
        }
    }
    vector<int> Stack;
    vector<int> path;
    int m, ver;
    Stack.push_back(1);
    while (!Stack.empty()) {
        m = 0;
        ver = Stack[Stack.size() - 1];
        for (int i = 0; i < v; i++) {
            if (graph[ver - 1][i]) {
                m = i + 1;
                graph[ver - 1][i] = 0;
                graph[i][ver - 1] = 0;
                Stack.push_back(m);
                break;
            }
        }
        if (m == 0) {
            path.push_back(ver);
            Stack.pop_back();
        }
    }
    for (int i = path.size() - 1; i > 0; i--) {
        cout << path[i] << "->";
    }
    cout << path[0];

    return 0;
}
```

```
Quantity :11
Matrix:
0 1 0 0 0 0 1 0 0 0 0
1 0 1 0 0 1 0 0 0 0 1
0 1 0 0 1 1 0 0 0 0 1
0 0 0 0 1 0 0 0 0 0 1
0 0 1 1 0 1 0 1 0 0 0
0 1 1 0 1 0 1 1 1 1 1
1 0 0 0 0 1 0 0 1 0 1
0 0 0 0 1 1 0 0 1 1 0
0 0 0 0 0 1 1 1 0 1 0
0 0 0 0 0 1 0 1 1 0 1
0 1 1 1 0 1 1 0 0 1 0
1->2->3->5->4->11->2->6->3->11->6->5
->8->6->7->9->6->10->8->9->10->11->7->1
```

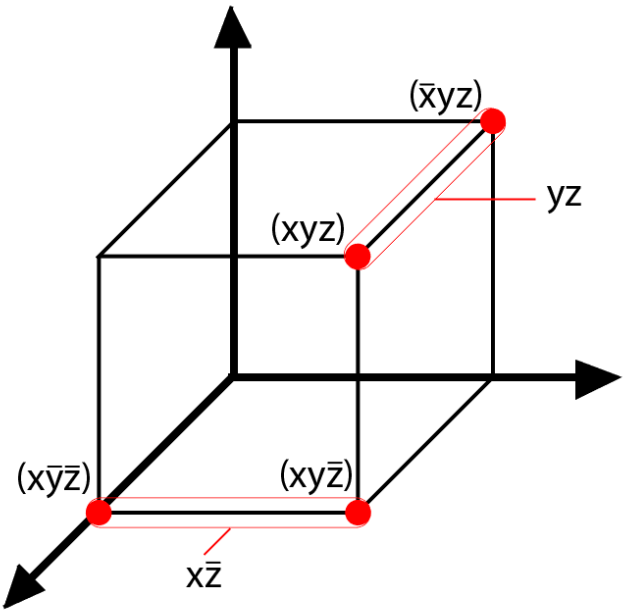
Завдання №9

Спростити формули (привести їх до скороченої ДНФ).

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

x	y	z	$\bar{z}$	xy	yz	$x\bar{z}$	$x\bar{z} \vee xy$	*	
0	0	0	1	0	0	0	0	0	
0	0	1	0	0	0	0	0	0	
0	1	0	1	0	0	0	0	0	
0	1	1	0	0	1	0	0	1	$\bar{x}yz$
1	0	0	1	0	0	1	1	1	$x\bar{y}\bar{z}$
1	0	1	0	0	0	0	0	0	
1	1	0	1	1	0	1	1	1	$xy\bar{z}$
1	1	1	0	1	1	0	1	1	xyz

$x\bar{y}\bar{z} \vee \bar{x}yz \vee xy\bar{z} \vee xyz$



СДНФ

$x\bar{z} \vee yz$