

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

Кафедра систем штучного інтелекту

Розрахункова робота
з дисципліни
«Дискретна математика»

Виконала:

студентка групи КН-112

Горішна Надя

Викладач:

Мельникова Н.І.

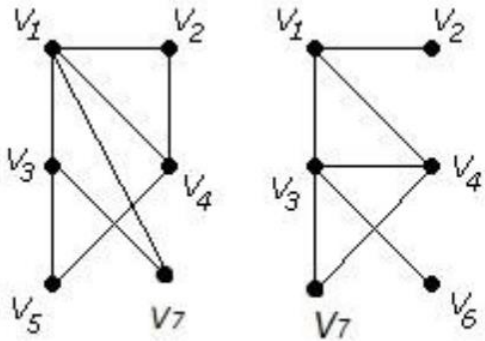
Львів-2019

Варіант 9

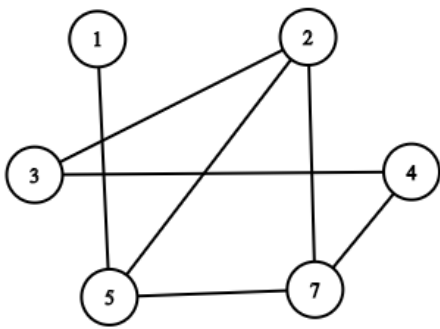
Завдання № 1

Виконати наступні операції над графами:

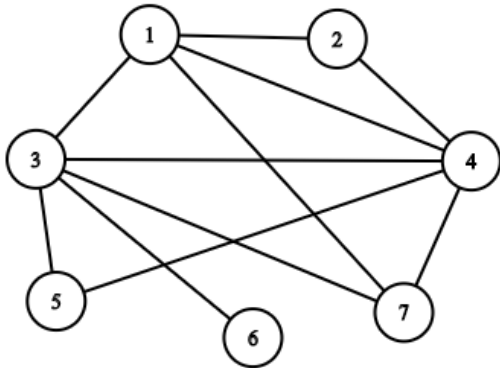
9)



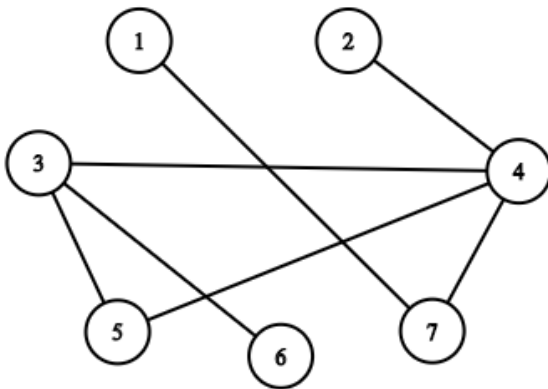
1) знайти доповнення до першого графу



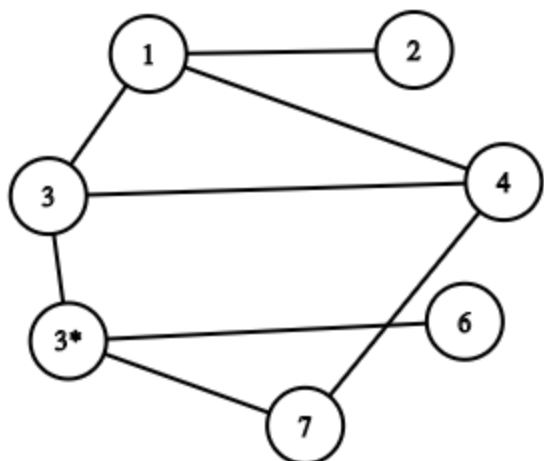
2) об'єднання графів



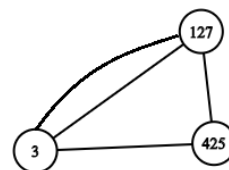
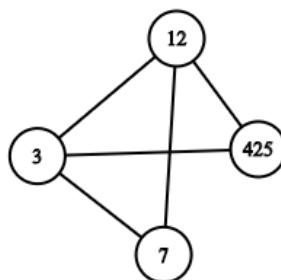
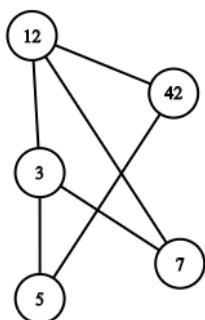
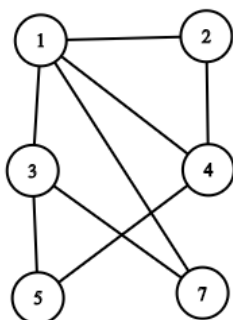
3) кільцеву суму $G1$ та $G2$ ($G1+G2$)



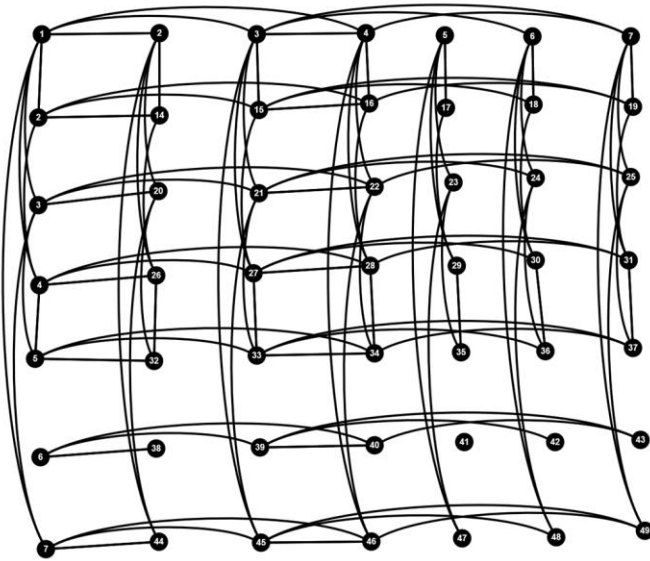
4) розмножити вершину у другому графі



5) виділити підграф А - що складається з 3-х вершин в G1



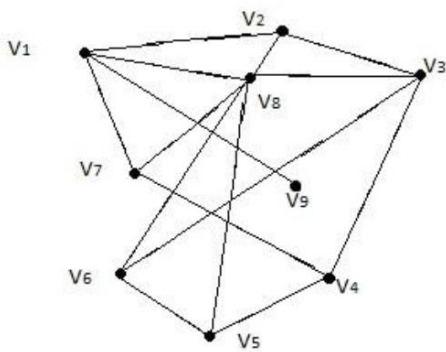
6) добуток графів



Завдання № 2

Скласти таблицю суміжності для орграфа.

9)



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

Завдання № 3

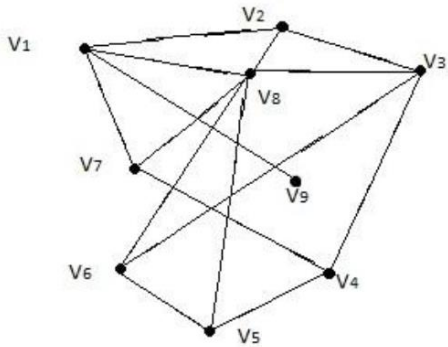
Для графа з другого завдання знайти діаметр.

Діаметр = 3

Завдання № 4

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

9)



Обхід графа вглиб:

1. Починаємо з довільної вершини і включаємо її в стек.
2. Розглядаємо останню вершину у стеку. Вибираємо інцидентне ребро, яке з'єднує поточну вершину із сусідньою, якщо її нема у стеку, і включаємо її туди.
3. Повторюємо пункт 2 поки остання вершина у стеку не матиме інцидентних ребер, тоді переходимо до пункту 4.
4. Виключаємо останню вершину із стеку і повертаємося до кроку 3. Якщо стек пустий, то обхід дерева завершено.

Вершина	Вміст стеку
V1	V1
V2	V1V2
V3	V1V2V3
V4	V1V2V3V4
V5	V1V2V3V4V5
V6	V1V2V3V4V5V6
V8	V1V2V3V4V5V6V8

V7	V1V2V3V4V5V6V8V7
-	V1V2V3V4V5V6V8
-	V1V2V3V4V5V6
-	V1V2V3V4V5
-	V1V2V3V4
-	V1V2V3
-	V1V2
-	V1
V9	V1V9
-	V1
-	Ø

Програмна реалізація:

```

1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int counter = 0;
5 bool check(vector<int> V, int Node) {
6     for (auto i = V.begin(); i != V.end(); i++)
7         if (*i == Node) return false;
8     return true;
9 }
10 void vhlub(vector<int>* V, int** arr, int n, int pos, vector<int>* WAS) {
11     for (int i = pos, k = 0; k < 1; i++, k++) {
12         for (int j = 0; j < n; j++)
13             if (arr[i][j] == 1 && check(*WAS, j)) {
14                 (*V).push_back(j);
15                 (*WAS).push_back(j);
16                 for (auto it = (*V).begin(); it != (*V).end(); it++)
17                     cout << *it + 1 << " ";
18                 cout << endl;
19                 vhlub(V, arr, n, j, WAS);
20             }
21     }
22     for (auto it = (*V).begin(); it != (*V).end(); it++)
23         cout << *it + 1 << " ";
24     cout << endl;
25     V->erase(V->end() - 1);
26     counter++;
27 }
28 int main() {
29     int n;
30     cout << "Enter number of nodes: ";
31     cin >> n;
32     int** arr = new int* [n];
33     for (int i = 0; i < n; i++)
34         arr[i] = new int[n];
35     for (int i = 0; i < n; i++)
36         for (int j = 0; j < n; j++)
37             cin >> arr[i][j];
38     vector<int> V;
39     vector<int> WAS;
40     V.push_back(0);
41     WAS.push_back(0);
42     cout << V[0] + 1 << endl;
43     vhlub(&V, arr, n, 0, &WAS);
44     for (auto it = V.begin(); it != V.end(); it++)
45         cout << *it + 1 << " ";
46     return 0;
47 }

```

```

22     for (auto it = (*V).begin(); it != (*V).end(); it++)
23         cout << *it + 1 << " ";
24     cout << endl;
25     V->erase(V->end() - 1);
26     counter++;
27 }
28 int main() {
29     int n;
30     cout << "Enter number of nodes: ";
31     cin >> n;
32     int** arr = new int* [n];
33     for (int i = 0; i < n; i++)
34         arr[i] = new int[n];
35     for (int i = 0; i < n; i++)
36         for (int j = 0; j < n; j++)
37             cin >> arr[i][j];
38     vector<int> V;
39     vector<int> WAS;
40     V.push_back(0);
41     WAS.push_back(0);
42     cout << V[0] + 1 << endl;
43     vhlub(&V, arr, n, 0, &WAS);
44     for (auto it = V.begin(); it != V.end(); it++)
45         cout << *it + 1 << " ";
46     return 0;
47 }

```

Результат:

```

for (auto it = (*i).begin(); it != (*i).end(); ++it)
{
    Console.WriteLine(it.ToString());
}
}

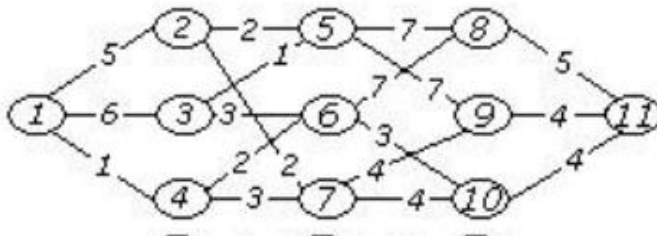
Enter number of nodes: 9
0 1 0 0 0 0 1 1 1
1 0 1 0 0 0 0 1 0
0 1 0 1 0 1 0 1 0
0 0 0 1 0 1 0 1 0
0 0 0 1 0 1 0 1 0
0 0 0 1 0 1 0 1 0
0 0 0 1 0 1 0 1 0
1 0 0 1 0 0 0 1 0
1 1 1 0 1 1 1 0 0
1 1 0 0 0 0 0 0 0
1 1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 8
1 2 3 4 5 6 8 7
1 2 3 4 5 6 8 7
1 2 3 4 5 6 8
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1 2
1 9
1 9
1
C:\Users\Horishni\source\repos\?
????? ? ????? 0.
????? ???????? ??? ?????, ????????

```

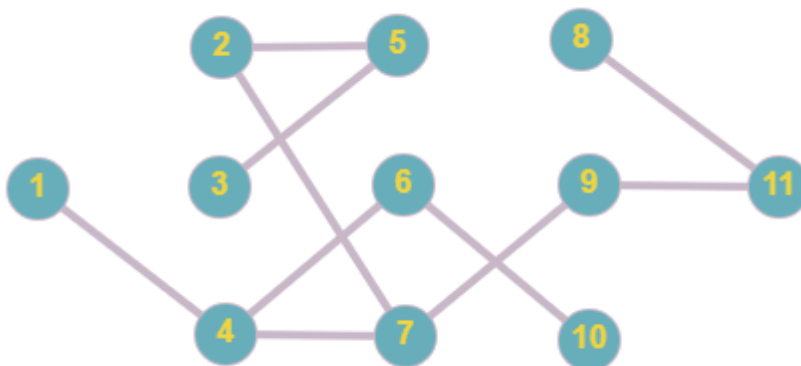
Завдання № 5

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

9)



Прима:



$V = \{1, 4, 6, 7, 2, 5, 3, 10, 9, 11, 8\}$

$E = \{(1, 4), (4, 6), (4, 7), (7, 2), (2, 5), (5, 3), (6, 10), (7, 9), (9, 11), (11, 8)\}$

Програмна реалізація:

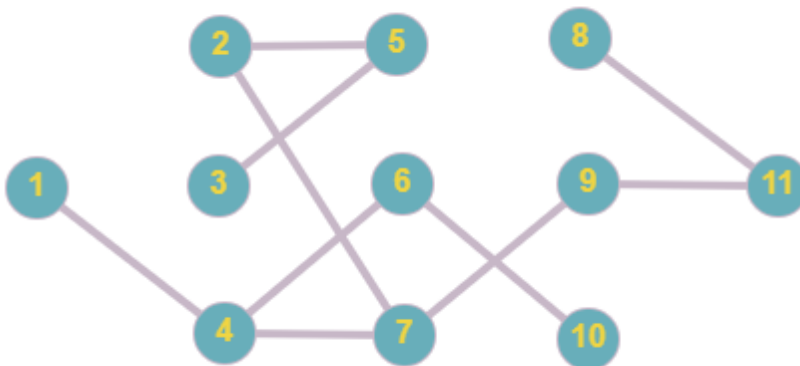
```
скр41.cpp
[дискр41] (Глобальная область)
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int vershunu, counter = 0, min = 0, k, t;
5      bool check = false;
6      cout << "Enter quantity of vertexes : ";
7      cin >> vershunu;
8      int* V = new int[vershunu];
9      int** graph = new int* [vershunu];
10     int** rebr = new int* [vershunu - 1];
11
12     for (int j = 0; j < vershunu; j++) {
13         graph[j] = new int[vershunu];
14     }
15     for (int j = 0; j < vershunu - 1; j++) {
16         rebr[j] = new int[2];
17     }
18     for (int i = 0; i < vershunu; i++) {
19         for (int j = 0; j < vershunu; j++) {
20             cin >> graph[i][j];
21         }
22     }
23     V[counter] = 1;
24     counter++;
25     for (int i = 0; counter < vershunu; i++) {
26         for (int j = 0; j < counter; j++) {
27             for (int a = 0; a < vershunu; a++) {
28                 for (int m = 0; m < counter; m++) {
29                     if (V[m] == a + 1) {
30                         check = true;
31                     }
32                 }
33             }
34             if (check) {
```

```
34         check = false;
35         continue;
36     }
37     if (min == 0 && graph[V[j] - 1][a] > 0) {
38         min = graph[V[j] - 1][a];
39         k = rebr[counter - 1][0] = V[j];
40         t = rebr[counter - 1][1] = a + 1;
41         continue;
42     }
43     if (graph[V[j] - 1][a] > 0 && graph[V[j] - 1][a] < min) {
44         min = graph[V[j] - 1][a];
45         k = rebr[counter - 1][0] = V[j];
46         t = rebr[counter - 1][1] = a + 1;
47     }
48 }
49
50 graph[k - 1][t - 1] = 0;
51 graph[t - 1][k - 1] = 0;
52
53 V[counter] = t;
54 counter++;
55 min = 0;
56 }
57
58 cout << "V: ";
59 for (int j = 0; j < vershunu; j++) {
60     cout << V[j] << " ";
61 }
62 cout << endl << "E: ";
63 for (int j = 0; j < vershunu - 1; j++) {
64     cout << "(" << rebr[j][0] << " ", << rebr[j][1] << " ), ";
65 }
66 delete[] V;
```

Результат програми:

```
Консоль отладки Microsoft Visual Studio
Enter quantity of vertexes : 11
0 5 6 1 0 0 0 0 0 0 0
5 0 0 0 2 0 2 0 0 0 0
6 0 0 0 1 3 0 0 0 0 0
1 0 0 0 0 2 3 0 0 0 0
0 2 1 0 0 0 0 7 7 0 0
0 0 3 2 0 0 0 7 0 3 0
0 2 0 3 0 0 0 0 4 4 0
0 0 0 0 7 7 0 0 0 5
0 0 0 0 7 0 4 0 0 0 4
0 0 0 0 0 3 4 0 0 0 4
0 0 0 0 0 0 0 5 4 4 0
U: 1, 4, 6, 7, 2, 5, 3, 10, 9, 11, 8,
E: < 1, 4 >, < 4, 6 >, < 4, 7 >, < 7, 2 >, < 2, 5 >, < 5, 3 >, < 6, 10 >, < 7, 9 >,
< 10, 11 >, < 11, 8 >,
C:\Users\Horishni\source\repos\????4.1\Debug\????4.1.exe <?????? 8752> ??????
??? ?????? ? ?????? -1073741819.
????? ??????? ???? ?????, ??????? ?????? ???????.
```

Краскала:



$V=\{1,4,3,5,6,2,7,10,9,11,8\}$

$E=\{(1,4),(3,5),(4,6),(5,2),(2,7),(6,10),(7,9),(9,11),(11,8)\}$

Програмна реалізація:

```
кр4.cpp - (Глобальная область)
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 const int SIZE = 11;
8
9 struct rebro
10 {
11     int e1;
12     int e2;
13     int weight;
14 };
15
16 bool operator < (const rebro& rebro1, const rebro& rebro2)
17 {
18     return rebro2.weight < rebro1.weight;
19 }
20
21 bool checking_top(vector<int> mas, int top)
22 {
23     for (int i = 0; i < mas.size(); i++)
24     {
25         if (mas[i] == top) return true;
26     }
27     return false;
28 }
29
30 bool checking_loop(vector<int> mas, int top1, int top2)
31 {
32     bool res_top1 = false, res_top2 = false;
33     for (int i = 0; i < mas.size(); i++)
```

```
кр4.cpp - (Глобальная область)
34     {
35         if (mas[i] == top1) res_top1 = true;
36         if (mas[i] == top2) res_top2 = true;
37     }
38     return res_top1 && res_top2;
39 }
40
41 bool trees_unit(vector<int> trees[SIZE], int first, int second)
42 {
43     int i, j, first_tree = 0, second_tree = 0;
44
45     for (i = 0; i < SIZE; i++)
46     {
47         if (trees[i].size() != 0)
48         {
49             for (j = 0; j < trees[i].size(); j++)
50             {
51                 if (trees[i][j] == first) first_tree = i;
52                 if (trees[i][j] == second) second_tree = i;
53             }
54         }
55     }
56
57     for (i = 0; i < trees[second_tree].size(); i++) trees[first_tree].push_back(trees[second_tree][i]);
58     trees[second_tree].clear();
59     return 0;
60 }
61
62 int main()
63 {
64     vector<rebro> Graf, Graf_MOT;
65     vector<int> trees[SIZE];
66 }
```

```

диск 4.cpp (Глобальная область)
67
68 int Matrix[SIZE][SIZE] = {
69     {0,5,6,1,0,0,0,0,0,0},
70     {5,0,0,2,0,2,0,0,0,0},
71     {6,0,0,0,1,3,0,0,0,0},
72     {1,0,0,0,0,2,3,0,0,0},
73     {0,2,1,0,0,0,0,7,7,0},
74     {0,0,3,2,0,0,0,7,0,3},
75     {0,2,0,3,0,0,0,4,4,0},
76     {0,0,0,7,7,0,0,0,0,5},
77     {0,0,0,7,0,4,0,0,0,4},
78     {0,0,0,0,3,4,0,0,0,4},
79     {0,0,0,0,0,0,5,4,4,0}
80 };
81
82 for (int i = 0; i < SIZE; i++)
83 {
84     for (int j = i; j < SIZE; j++)
85     {
86         if (Matrix[i][j] != 0)
87         {
88             rebro help;
89             cout << Matrix[i][j] << "(" << i + 1 << ", " << j + 1 << ")" << endl;
90             help.e1 = i;
91             help.e2 = j;
92             help.weight = Matrix[i][j];
93             Graf.push_back(help);
94         }
95     }
96 }
97
98 cout << "-----" << endl;
99
100 % Проблемы не найдены.

```

```

диск 4.cpp (Глобальная область)
sort(Graf.rbegin(), Graf.rend());
for (int i = 0; i < Graf.size(); i++) cout << Graf[i].weight << "(" << Graf[i].e1 + 1 << ", " << Graf[i].e2 + 1 << ")" << endl;

cout << "-----" << endl;

for (int i = 0; i < SIZE; i++) trees[i].push_back(i);

for (int i = 0; i < Graf.size(); i++)
{
    for (int j = 0; j < SIZE; j++)
    {
        bool res_top1 = checking_top(trees[j], Graf[i].e1);
        bool res_top2 = checking_top(trees[j], Graf[i].e2);
        bool res_loop = !checking_loop(trees[j], Graf[i].e1, Graf[i].e2);

        if ((res_top1 || res_top2) && res_loop)
        {
            trees_unit(trees, Graf[i].e1, Graf[i].e2);

            rebro help;
            help.e1 = Graf[i].e1;
            help.e2 = Graf[i].e2;
            help.weight = Graf[i].weight;
            Graf_MOT.push_back(help);

            break;
        }
    }
}

for (int i = 0; i < Graf_MOT.size(); i++) cout << Graf_MOT[i].weight << "(" << Graf_MOT[i].e1 + 1 << ", " << Graf_MOT[i].e2 + 1 << ")" << endl;

```

Результат:

```

7<6,8>
-----
1<1,4>
1<3,5>
2<2,5>
2<2,7>
2<4,6>
3<3,6>
3<6,10>
4<7,9>
4<9,11>
5<8,11>

C:\Users\Horishni\source\repos\???? 4\Debug\???? 4.exe <?????? 9460> ???????
? ?????? ? ?????? 0.
????? ??????? ??? ?????, ??????? ?????? ???????

```

Завдання № 6

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

9)

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

1-8-5-4-7-2-6-3-1(21)

1-8-7-4-5-2-6-3-1(19)

2-6-8-1-4-5-3-7-2(23)

2-6-8-1-5-4-7-3-2(19)

3-2-6-8-1-5-4-7-3(19)

4-7-8-1-5-2-6-3-4(20)

5-4-7-8-1-6-2-3-5(19)

6-2-5-4-7-8-1-3-6(19)

7-4-5-2-6-8-1-3-7(19)

Найкоротші шляхи:

1-8-7-4-5-2-6-3-1(19)

2-6-8-1-5-4-7-3-2(19)

3-2-6-8-1-5-4-7-3(19)

4-5-8-1-6-2-3-7-4(19)

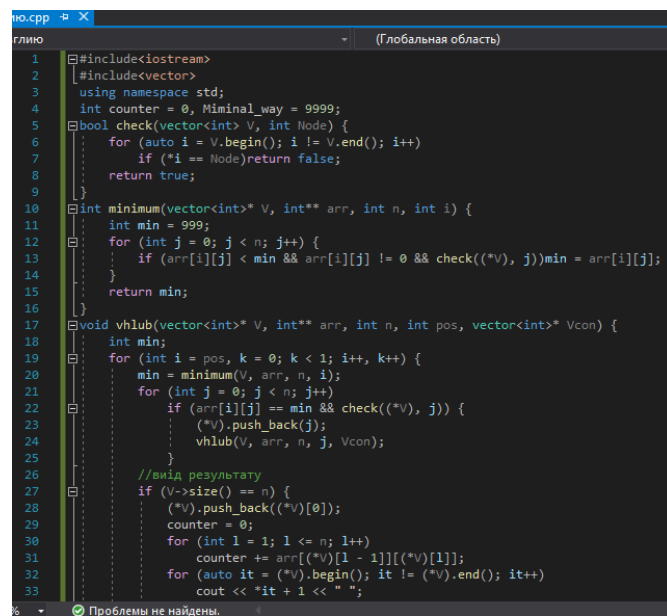
5-4-7-8-1-6-2-3-5(19)

6-2-5-4-7-8-1-3-6(19)

7-4-5-2-6-8-1-3-7(19)

7-4-5-8-1-6-2-3-7(19)

Програмна реалізація:



```
1 #include<iostream>
2 #include<vector>
3 using namespace std;
4 int counter = 0, Minimal_way = 9999;
5 bool check(vector<int> V, int Node) {
6     for (auto i = V.begin(); i != V.end(); i++)
7         if (*i == Node) return false;
8     return true;
9 }
10 int minimum(vector<int>* V, int** arr, int n, int i) {
11     int min = 999;
12     for (int j = 0; j < n; j++) {
13         if (arr[i][j] < min && arr[i][j] != 0 && check(*V, j)) min = arr[i][j];
14     }
15     return min;
16 }
17 void vhlub(vector<int>* V, int** arr, int n, int pos, vector<int>* Vcon) {
18     int min;
19     for (int i = pos, k = 0; k < 1; i++, k++) {
20         min = minimum(V, arr, n, i);
21         for (int j = 0; j < n; j++)
22             if (arr[i][j] == min && check(*V, j)) {
23                 (*V).push_back(j);
24                 vhlub(V, arr, n, j, Vcon);
25             }
26         //вивід результату
27         if (V->size() == n) {
28             (*V).push_back((*V)[0]);
29             counter = 0;
30             for (int l = 1; l <= n; l++)
31                 counter += arr[(*V)[l-1]][(*V)[l]];
32             for (auto it = (*V).begin(); it != (*V).end(); it++)
33                 cout << *it + 1 << " ";
34         }
35     }
36 }
```

```
1.cpp -# X
(Глобальная область)
34 cout << " (" << counter << ")" << endl;
35 if (Minimal_way == counter) {
36     for (int op = 0; op <= n; op++)
37         (*Vcon).push_back((*V)[op]);
38     (*Vcon).push_back(counter);
39 }
40 else if (Minimal_way > counter) {
41     (*Vcon).clear();
42     for (int op = 0; op <= n; op++)
43         (*Vcon).push_back((*V)[op]);
44     (*Vcon).push_back(counter);
45     Minimal_way = counter;
46 }
47 V->pop_back();
48 }
49 }
50 V->pop_back();
51 }
52 int main()
53 {
54     int n;
55     cout << "Enter number of nodes: ";
56     cin >> n;
57     int** arr = new int* [n];
58     for (int i = 0; i < n; i++)
59         arr[i] = new int[n];
60     for (int i = 0; i < n; i++)
61         for (int j = 0; j < n; j++)
62             cin >> arr[i][j];
63     vector<int> V;
64     vector<int> Vcon;
65     cout << endl;
66     for (int i = 0; i < n; i++) {
```

```
100.cpp -# X
(Глобальная область)
55 cout << "Enter number of nodes: ";
56 cin >> n;
57 int** arr = new int* [n];
58 for (int i = 0; i < n; i++)
59     arr[i] = new int[n];
60 for (int i = 0; i < n; i++)
61     for (int j = 0; j < n; j++)
62         cin >> arr[i][j];
63 vector<int> V;
64 vector<int> Vcon;
65 cout << endl;
66 for (int i = 0; i < n; i++) {
67     V.clear();
68     V.push_back(i);
69     vhlub(&V, arr, n, i, &Vcon);
70 }
71 cout << endl << endl << endl << "rez" << endl << endl;
72 for (int i = 1; i <= Vcon.size(); i++) {
73     if (i != 0 && i % (n + 2) == 0)
74         cout << " (" << Vcon[i - 1] << ")" << endl;
75     else
76         cout << Vcon[i - 1] + 1 << " ";
77 }
78 }
```

Результат:

```

Консоль отладки Microsoft Visual Studio
Enter number of nodes: 8
0 5 5 3 3 4 4 1
5 0 4 3 2 1 4 1
5 4 0 4 5 6 5 5
3 3 4 0 1 5 1 7
3 2 5 1 0 5 5 2
4 1 6 5 5 0 7 3
4 4 5 1 5 7 0 2
1 6 5 7 2 3 2 0

1 8 5 4 7 2 6 3 1 (21)
1 8 7 4 5 2 6 3 1 (19)
2 6 8 1 4 5 3 3 2 (23)
2 6 8 1 4 7 3 5 2 (21)
2 6 8 1 4 7 5 3 2 (23)
2 6 8 1 5 4 7 3 2 (19)
3 2 6 8 1 4 5 7 3 (23)
3 2 6 8 1 4 7 5 3 (23)
3 2 6 8 1 5 4 7 3 (19)
3 4 5 2 6 8 1 7 3 (21)
3 4 5 8 1 6 2 7 3 (22)
3 4 5 8 1 7 2 6 3 (23)
3 4 7 8 1 5 2 6 3 (20)
4 5 2 6 8 1 7 3 4 (21)
4 5 8 1 6 2 7 3 4 (19)
4 5 8 1 6 2 7 3 4 (22)
4 5 8 1 7 2 6 3 4 (23)
4 7 8 1 5 2 6 3 4 (20)
5 4 7 8 1 6 2 3 5 (19)
6 2 5 4 7 8 1 3 6 (19)
7 4 5 2 6 8 1 3 7 (19)
7 4 5 8 1 6 2 3 7 (19)
8 1 4 5 2 6 3 7 8 (21)
8 1 4 7 2 6 5 3 8 (25)
8 1 5 4 7 2 6 3 8 (22)

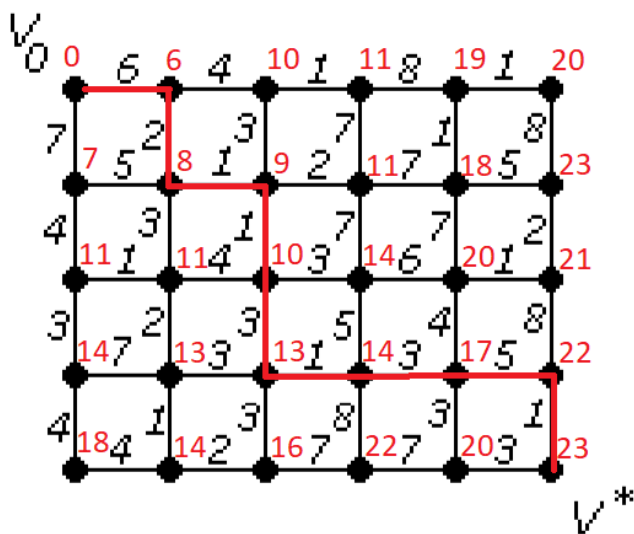
rez
1 8 7 4 5 2 6 3 1 (19)
2 6 8 1 5 4 7 3 2 (19)
3 2 6 8 1 5 4 7 3 (19)
4 5 8 1 6 2 3 7 4 (19)
5 4 7 8 1 6 2 3 5 (19)
6 2 5 4 7 8 1 3 6 (19)
7 4 5 2 6 8 1 3 7 (19)
7 4 5 8 1 6 2 3 7 (19)

C:\Users\Horishni\source\repos\?????\Debug\?????.
???? ? ????? 0.
????? ?????? 222 2222 222222 222222 222222

```

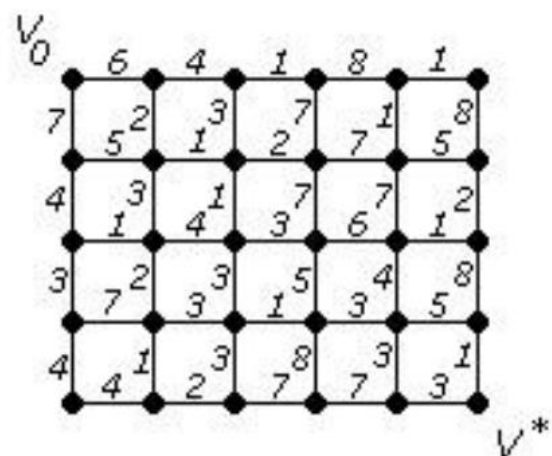
Завдання № 7

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



9)

1



Програмна реалізація:

```
кр5.2.cpp
дискр5.2 (Глобальная область)
1  #include <iostream>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #define SIZE 30
5  #define _CRT_SECURE_NO_WARNINGS
6  #pragma warning(disable : 4996);
7  using namespace std;
8  int main()
9  {
10     int a[SIZE][SIZE];
11     int d[SIZE];
12     int v[SIZE];
13     int t, minindex, min;
14     int first = 0;
15     cout << "enter matrix :" << endl;
16     for (int i = 0; i < SIZE; i++) {
17         for (int j = 0; j < SIZE; j++) {
18             cin >> a[i][j];
19         }
20     }
21     for (int i = 0; i < SIZE; i++)
22     {
23         d[i] = 10000;
24         v[i] = 1;
25     }
26     d[first] = 0;
27     do {
28         minindex = 10000;
29         min = 10000;
30         for (int i = 0; i < SIZE; i++)
31         {
32             if ((v[i] == 1) && (d[i] < min))
33             {
```

```
кр5.2.cpp
дискр5.2 (Глобальная область)
34         min = d[i];
35         minindex = i;
36     }
37     }
38     if (minindex != 10000)
39     {
40         for (int i = 0; i < SIZE; i++)
41         {
42             if (a[minindex][i] > 0)
43             {
44                 t = min + a[minindex][i];
45                 if (t < d[i])
46                 {
47                     d[i] = t;
48                 }
49             }
50         }
51         v[minindex] = 0;
52     }
53     } while (minindex < 10000);
54     cout << "It's the shortest distance to the vertexes" << endl;
55     for (int i = 0; i < SIZE; i++)
56         cout << d[i] << " ";
57     int ver[SIZE];
58     int end = 29;
59     ver[0] = end + 1;
60     int k = 1;
61     int weight = d[end];
62
63     while (end != first)
64     {
65         for (int i = 0; i < SIZE; i++)
66             if (a[end][i] != 0)
```

```

дискр5.2 (Глобальная область)
55 for (int i = 0; i < SIZE; i++)
56     cout << d[i] << " ";
57 int ver[SIZE];
58 int end = 29;
59 ver[0] = end + 1;
60 int k = 1;
61 int weight = d[end];
62
63 while (end != first)
64 {
65     for (int i = 0; i < SIZE; i++)
66         if (a[end][i] != 0)
67         {
68             int t = weight - a[end][i];
69             if (t == d[i])
70             {
71                 weight = t;
72                 end = i;
73                 ver[k] = i + 1;
74                 k++;
75             }
76         }
77     cout << endl;
78     cout << "your the shortest way" << endl;
79     for (int i = k - 1; i >= 0; i--) {
80         cout << "V" << ver[i] << " -- ";
81     }
82     getch(); getch();
83     return 0;
84 }
85
86
87

```

Результат:

```

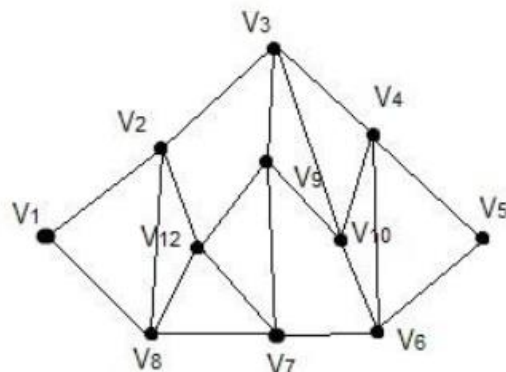
C:\Windows\system32\cmd.exe
0 0 0 0 0 0 4 0 0 0 0 0 0 1 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 4 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 4 0 3 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 3 0 6 0 0 0 0 5 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 6 0 1 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 2 0 0 0 0 0 1 0 0 0 0 0 8 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 0 0 7 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 2 0 0 0 0 0 7 0 3 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 3 0 0 0 0 3 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 5 0 0 0 0 1 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 0 0 4 0 0 0 0 0 3 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 0 0 8 0 0 0 0 0 5 0 0 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 4 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 0 0 1 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 0 0 0 0 3 0 0 0 0 2 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 0 0 8 0 0 0 0 7 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 0 0 0 0 3 0 0 0 0 7 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 0 0 0 1 0 0 0 0 3 0 0 0
It's the shortest distance to the vertexes
0 6 10 11 19 20 7 8 9 11 18 22 11 11 10 13 19 20 14 13 13 1
4 17 22 18 14 16 22 20 23
your the shortest way
01 -- 02 -- 08 -- 09 -- 015 -- 021 -- 022 -- 023 -- 024 -- 030 --

```

Завдання № 8

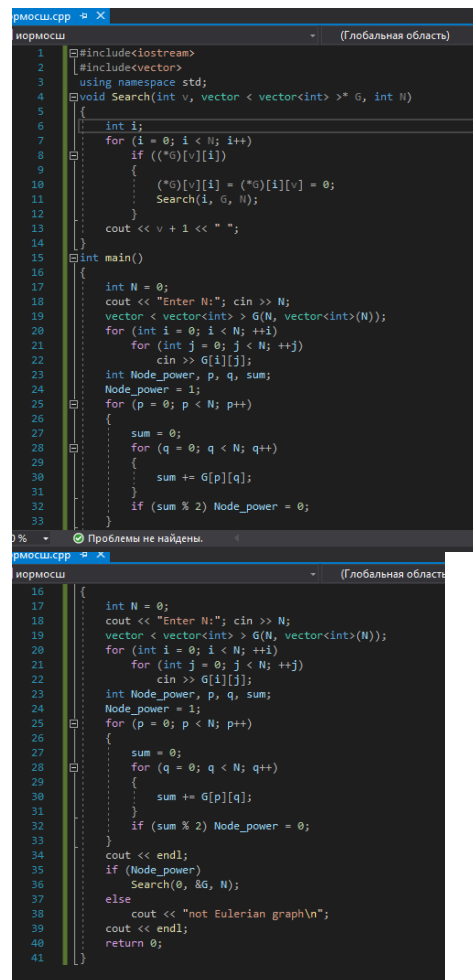
Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.

Алгоритм Флері(якщо степені вершин парні):



1. Вибираємо почтакову вершину і починаємо рух по інцидентному ребру.
 2. Повторюємо пункт 1, вважаючи вершину в яку прибули початковою.
 3. Не проходимо по ребру, якщо видалення цього ребра призводить до розбиття графа.
 4. Робота закінчена, коли ми пройшли по усіх ребрах
- а) 1-8-11-7-8-2-11-9-10-6-7-9-3-10-4-6-5-4-3-2-1

Програмна реалізація:



```
1 #include<iostream>
2 #include<vector>
3 using namespace std;
4 void Search(int v, vector< vector<int> >* G, int N)
5 {
6     int i;
7     for (i = 0; i < N; i++)
8     {
9         if ((*G)[v][i])
10         {
11             (*G)[v][i] = (*G)[i][v] = 0;
12             Search(i, G, N);
13         }
14     }
15     cout << v + 1 << " ";
16 }
17 int main()
18 {
19     int N = 0;
20     cout << "Enter N: "; cin >> N;
21     vector< vector<int> > G(N, vector<int>(N));
22     for (int i = 0; i < N; ++i)
23     {
24         for (int j = 0; j < N; ++j)
25             cin >> G[i][j];
26     }
27     int Node_power, p, q, sum;
28     Node_power = 1;
29     for (p = 0; p < N; p++)
30     {
31         sum = 0;
32         for (q = 0; q < N; q++)
33         {
34             sum += G[p][q];
35         }
36         if (sum % 2) Node_power = 0;
37     }
38     cout << endl;
39     if (Node_power)
40     {
41         Search(0, &G, N);
42     }
43     else
44     {
45         cout << "not Eulerian graph\n";
46     }
47     cout << endl;
48     return 0;
49 }
```

Результат:

```

Консоль отладки Microsoft Visual Studio
int
for
Enter N:11
0 1 0 0 0 0 0 1 0 0 0
1 0 1 0 0 0 0 1 0 0 1
0 1 0 1 0 0 0 0 1 1 0
0 0 1 0 1 1 0 0 0 1 0
0 0 0 1 0 1 0 0 0 0 0
0 0 0 1 1 0 1 0 0 0 1
0 0 0 0 0 1 0 1 0 1 0
0 0 0 0 0 1 0 1 1 0 1
1 1 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 1 0 0 1 1
0 0 1 1 0 1 0 0 1 0 0
0 1 0 0 0 1 1 1 0 0
1 8 11 7 8 2 11 9 10 6 7 9 3 10 4 6 5 4 3 2 1
int
cout
vec
for
C:\Users\Horishni\source\repos\??????\Debug\??????
? ?????? ? ?????? 0.
????? ?????? ??? ?????, ?????? ?????? ???????.
int

```

Елементарних циклів:

Оскільки ейлеровів цикл складається з елементарних циклів, то ми шукаємо найкоротші, найефективніші цикли і об'єднуємо їх і в один.

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

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

Програмна реалізація:

```

нормосиш (Глобальная область)
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 vector<int> Vcon;
5 int Minimal_way = 999;
6 bool check(vector<int> V, int Node) {
7     for (auto i = V.begin(); i != V.end(); i++)
8         if (*i == Node) return false;
9     return true;
10 }
11 void vhlub(vector<int>* V, int** arr, int n, int pos, int start_node) {
12     for (int i = pos, k = 0; k < 1; i++, k++) {
13         for (int j = 0; j < n; j++)
14             if (arr[i][j] == 1 && check(*V, j)) {
15                 if (j == start_node && (*V).size() > 2) {
16                     if (Minimal_way > V->size()) {
17                         Vcon.clear();
18                         Vcon.push_back(start_node + 1);
19                         for (auto it = (*V).begin(); it != (*V).end(); it++)
20                             Vcon.push_back(*it + 1);
21                         Vcon.push_back(start_node + 1);
22                         Minimal_way = V->size();
23                         break;
24                     }
25                 }
26                 else {
27                     (*V).push_back(j);
28                     vhlub(V, arr, n, j, start_node);
29                 }
30             }
31     }
32     if (V->size() != 0)
33         V->pop_back();

```

```

нормосиш (Глобальная область)
34 }
35 int main() {
36     int n;
37     cout << "Enter number of nodes: ";
38     cin >> n;
39     int** arr = new int* [n];
40     for (int i = 0; i < n; i++)
41         arr[i] = new int[n];
42     for (int i = 0; i < n; i++)
43         for (int j = 0; j < n; j++)
44             cin >> arr[i][j];
45     vector<int> V;
46     vector<int> VAS;
47     cout << endl;
48     int Node_power, p, q, sum;
49     Node_power = 1;
50     for (p = 0; p < n; p++)
51     {
52         sum = 0;
53         for (q = 0; q < n; q++)
54         {
55             sum += arr[p][q];
56         }
57         if (sum % 2) Node_power = 0;
58     }
59     cout << endl;
60     if (Node_power) {
61         cout << "The shortest ways: " << endl;
62         for (int j = 0; j < n; j++) {
63             Minimal_way = 999;
64             vhlub(&V, arr, n, j, j);
65             for (int i = 1; i <= Vcon.size(); i++)
66                 cout << Vcon[i - 1] << " ";

```

```

нормосиш (Глобальная область)
52     sum = 0;
53     for (q = 0; q < n; q++)
54     {
55         sum += arr[p][q];
56     }
57     if (sum % 2) Node_power = 0;
58 }
59 cout << endl;
60 if (Node_power) {
61     cout << "The shortest ways: " << endl;
62     for (int j = 0; j < n; j++) {
63         Minimal_way = 999;
64         vhlub(&V, arr, n, j, j);
65         for (int i = 1; i <= Vcon.size(); i++)
66             cout << Vcon[i - 1] << " ";
67         cout << endl;
68         Vcon.clear();
69     }
70 }
71 else
72     cout << "not Eulerian graph\n";
73 cout << endl;
74 return 0;
75 }

```

Результат:

```

Консоль отладки Microsoft Visual Studio
Enter number of nodes: 11
0 1 0 0 0 0 0 1 0 0 0
1 0 1 0 0 0 0 1 0 0 1
0 1 0 1 0 0 0 0 1 1 0
0 0 1 0 1 1 0 0 0 1 0
0 0 0 1 0 1 0 0 0 0 0
0 0 0 1 1 0 1 0 0 1 0
0 0 0 0 0 1 0 1 1 0 1
1 1 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 1 0 0 1 1
0 0 1 1 0 1 0 0 1 0 0
0 1 0 0 0 0 1 1 1 0 0

The shortest ways:
1 2 11 8 1
2 1 8 11 2
3 2 11 9 3
4 3 9 10 4
5 4 10 6 5
6 4 3 10 6
7 6 10 9 7
8 1 2 11 8
9 3 2 11 9
10 3 4 6 10
11 2 1 8 11

C:\Users\Horishni\source\repos\??????\Debug
2 222222 2 22222 0

```

Завдання №9

Спростити формули (привести їх до скороченої ДНФ).

$$9. \quad (x \rightarrow y) \cdot (y \rightarrow z) \rightarrow (x \rightarrow z).$$

$$(\neg x \vee y)(\neg y \vee z) \rightarrow (x \rightarrow z)$$

$$(\neg x \neg y \vee \neg x z \vee y z) \rightarrow (x \rightarrow z)$$

$$(\neg x \neg y \vee y z) \rightarrow (\neg x \vee z)$$

$$\neg(\neg x \neg y \vee y z) \vee \neg x \vee z$$

$$((x \vee y)(\neg y \vee \neg z)) \vee \neg x \vee z$$

$$x \neg y \vee x z \vee y \neg y \vee y \neg z \vee \neg x \vee z$$

$$x \neg y \vee x z \vee y \neg z \vee \neg x \vee z$$

$$y \neg z \vee x \neg y \vee \neg x \vee z$$

$$\neg x \vee \neg y \vee \neg z \vee z$$

$$\neg x \vee \neg y \vee 1$$