

10.12.2019

**Кафедра систем штучного
інтелекту**

Розрахункова робота
з дисципліни
«Дискретна математика»

Виконав:

студент групи КН-113

Власюк Олександр

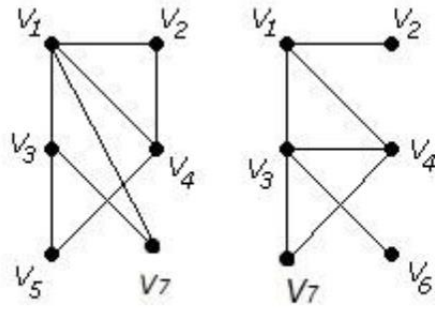
Викладач:

Мельникова Н.І.

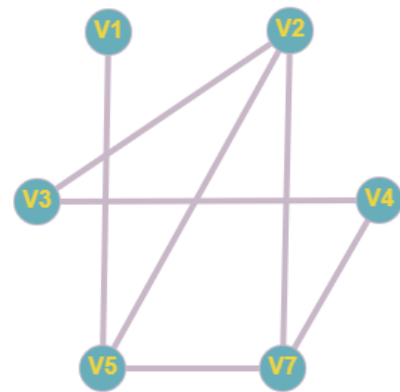
Варіант 9

Завдання № 1

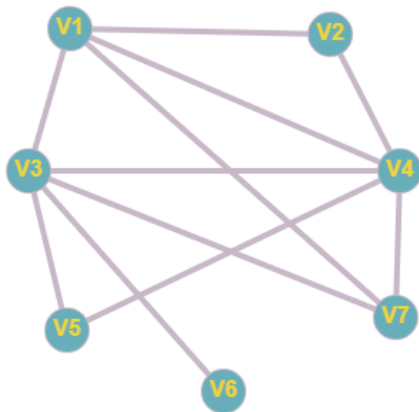
Виконати наступні операції над графами:



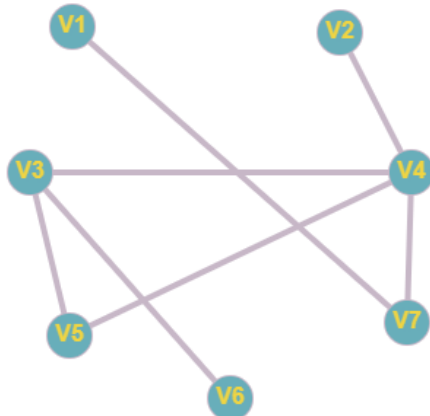
1) знайти доповнення до першого графу:



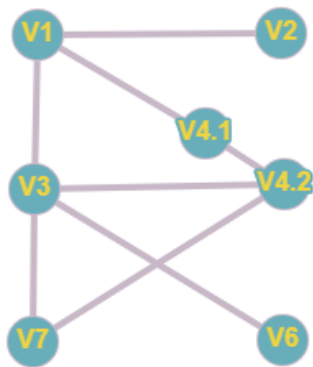
2) об'єднання графів:



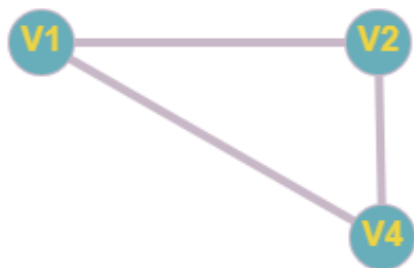
3) кільцеву суму G_1 та G_2 ($G_1 + G_2$):



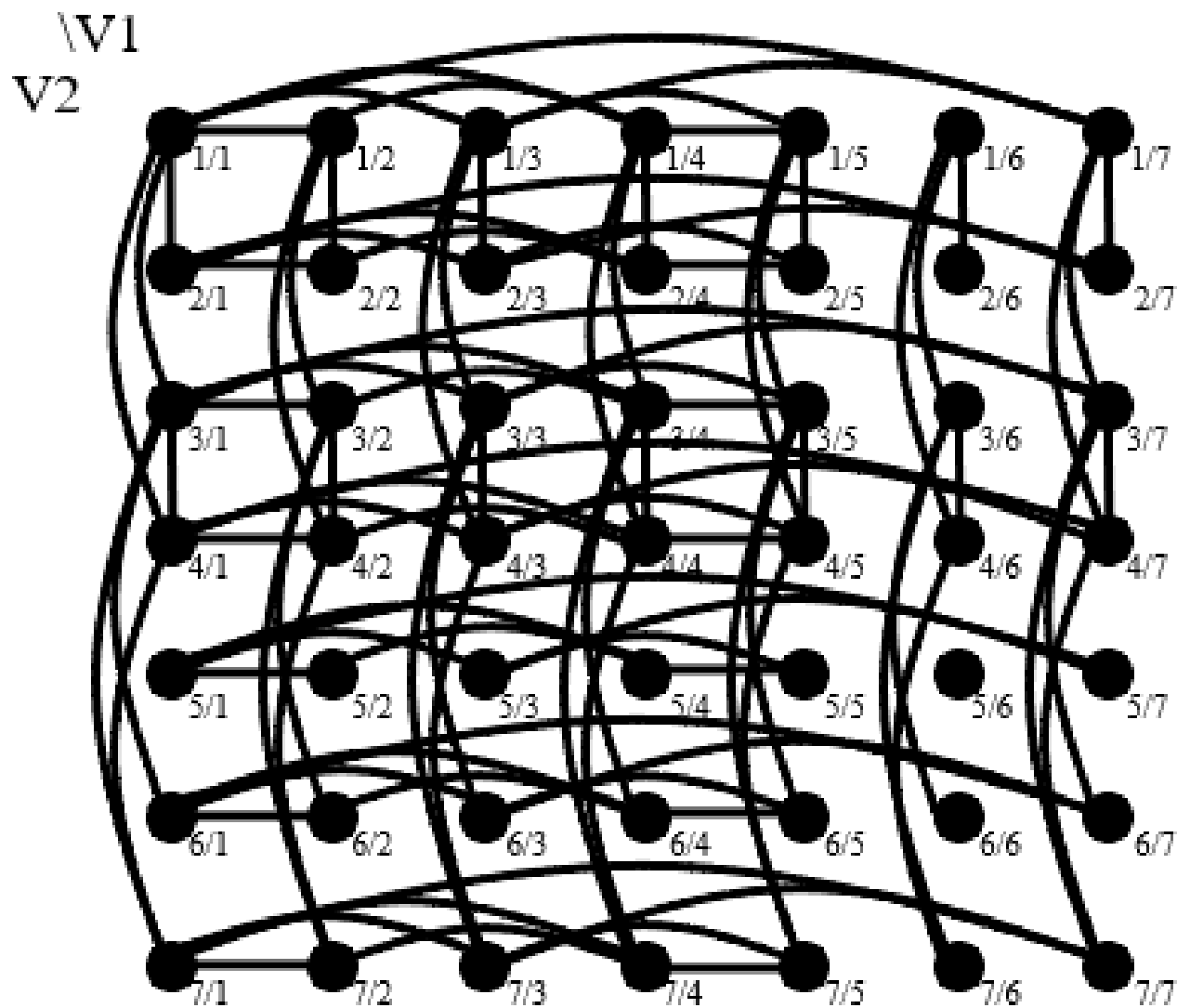
4) розмножити вершину у другому графі:



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

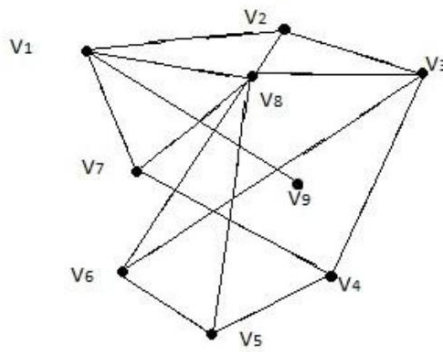


6) добуток графів:



Завдання № 2

Скласти таблицю суміжності для орграфа.



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

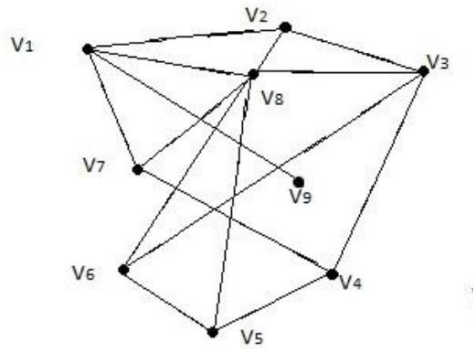
Завдання № 3

Для графа з другого завдання знайти діаметр.

Діаметр (3)

Завдання № 4

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

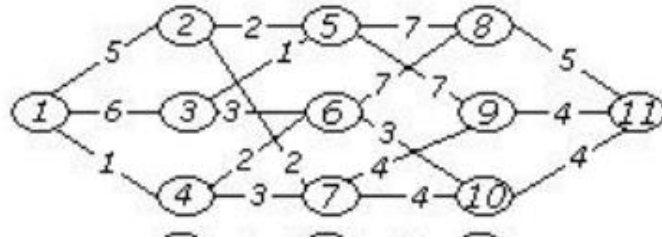


Обхід дерева вглиб:

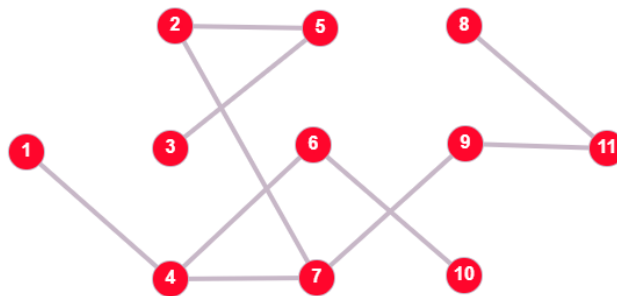
Вершина	Вміст стеку
V1	V1
V2	V1V2
V3	V1V2V3
V4	V1V2V3V4
V5	V1V2V3V4V5
V6	V1V2V3V4V5V6
V8	V1V2V3V4V5V6V8
V7	V1V2V3V4V5V6V8V7
-	V1V2V3V4V5V6V8
-	V1V2V3V4V5V6
-	V1V2V3V4V5
-	V1V2V3V4
-	V1V2V3
-	V1V2
-	V1
V9	V1V9
-	V1
-	∅
Кінець обходу	Кінець обходу

Завдання № 5

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



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:

```
1 #include <iostream>
2
3 using namespace std;
4
5 struct rebro
6 {
7     int t1;
8     int t2;
9     int waga;
10 };
11
12 void vvaid (rebro* p, int n, int m)
13 {
14     cout<<"Vvedite kbrki i yezshnya i yezshnya i waga:\n";
15     for (int i=0; i<m; i++)
16     {
17         cout<<"kbrki"<<i+1<<" ";
18         cin>>p[i].t1>>p[i].t2>>p[i].waga;
19         while (p[i].t1<0 || p[i].t1>n || p[i].t2<0 || p[i].t2>n || p[i].waga<0)
20         {
21             cout<<"Nepriyatni dani"<<endl;
22             cout<<"Vvedite dani znovo"<<endl;
23             cin>>p[i].t1>>p[i].t2>>p[i].waga;
24         }
25     }
26 }
27
28 void bulb (rebro* p, int n)
29 {
30     rebro temp;
31     for (int i=0; i<n; i++)
32     {
33         for (int j=0; j<n-i-1; j++)
34         {
35             if (p[j].waga > p[j+1].waga)
36             {
37                 temp = p[j];
38                 p[j] = p[j+1];
39                 p[j+1] = temp;
40             }
41         }
42     }
43 }
44
45 bool vkluchene (int* a, int n, int f)
46 {
47     for (int i=0; i<n; i++)
48     {
49         if (f == a[i])
50         {
51             return true;
52         }
53     }
54     return false;
55 }
56
57 bool minn (int w, rebro* ed, int m, int* v, int n)
58 {
59     for (int j=0; j<m; j++)
60     {
61         if (((!vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2)) ||
62             (vkluchene(v, n, ed[j].t1) && !vkluchene(v, n, ed[j].t2)))
63             && ed[j].waga < w)
64         {
65             return false;
66         }
67     }
68     return true;
69 }
70
71 void prima (rebro* ed, int* v, rebro* tree, int n, int m, int* i, int* j)
72 {
73     if (i==n)
74     {
75         return;
76     }
77     else if (j==m)
78     {
79         j=1;
80         vkluchene(v, n, ed[j].t2);
81     }
82 }
```

```

80     }
81     if(vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2))
82     {
83         j++;
84         pryma(ed, v, tree, n, m, i, j);
85     }
86     else if(!vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2))
87     {
88         tree[i-1]=ed[j];
89         v[i] = ed[j].t1;
90         j++;
91         i++;
92         pryma(ed, v, tree, n, m, i, j);
93     }
94     else if (vkluchene(v, n, ed[j].t1) && !vkluchene(v, n, ed[j].t2) && minn(ed[j].waga, ed, m, v, n))
95     {
96         tree[i-1]=ed[j];
97         v[i] = ed[j].t2;
98         j++;
99         i++;
100         pryma(ed, v, tree, n, m, i, j);
101     }
102     else
103     {
104         j++;
105         pryma(ed, v, tree, n, m, i, j);
106     }
107 }
108 }
109 }
110
111 int main()
112 {
113     int n,m;
114     cout<<"skilky vershyn hochete vvesty?";
115     cin>>n;
116     cout<<"skilky reber hochete vvesty?";
117     cin>>m;
118     cout<<endl;
119     rebro *ed = new rebro[m];
120
121     int *v = new int[n];
122     rebro *tree = new rebro [n-1];
123     v[0]=ed[0].t1;
124     v[1]=ed[0].t2;
125     tree[0]=ed[0];
126     int i=2;
127     int j=1;
128     pryma (ed,v,tree,n,m,i,j);
129     cout<<"\nV = { ";
130     for (int x=0; x<n; x++)
131     {
132         cout<<v[x]<<" ";
133     }
134     cout<<"\nE = { ";
135     for (int x=0; x<n-1; x++)
136     {
137         cout<<"{"<<tree[x].t1<<" "<<tree[x].t2<<" " ";
138     }
139     cout<<"\n";
140     return 0;
141 }
142
143

```

Результат:

```

skilky vershyn hochete vvesty?11
skilky reber hochete vvesty?18

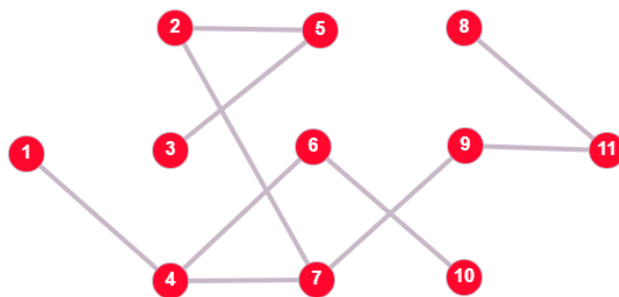
Vvedit' rebra(1 vershyna | 2vershyna | waga):
rebro1: 1 2 5
rebro2: 1 3 6
rebro3: 1 4 1
rebro4: 2 5 2
rebro5: 2 7 2
rebro6: 3 5 1
rebro7: 3 6 3
rebro8: 4 6 2
rebro9: 4 7 3
rebro10: 5 8 7
rebro11: 5 9 7
rebro12: 6 8 7
rebro13: 6 10 3
rebro14: 7 9 4
rebro15: 7 10 4
rebro16: 8 11 5
rebro17: 9 11 4
rebro18: 10 11 4

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) }

Process returned 0 (0x0)   execution time : 475.931 s
Press any key to continue.

```

Краскала:



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

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

Програмна реалізація:

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <iostream>
3  #include <string>
4  #include <cmath>
5  using namespace std;
6  FILE* f;
7
8  struct edge {
9      int v1;
10     int v2;
11     int w;
12 };
13
14 int findMin(edge g[], int n) {
15     int min = g[0].w;
16     for (int i = 0; i < n; i++) {
17         if (g[i].w < min) {
18             min = g[i].w;
19         }
20     }
21     return min;
22 }
23
24 int findMax(edge g[], int n) {
25     int max = g[0].w;
26     for (int i = 0; i < n; i++) {
27         if (g[i].w > max) {
28             max = g[i].w;
29         }
30     }
31     return max;
32 }
33
34 bool check(int v[], int n) {
35     for (int i = 0; i < 100; i++) {
36         if (v[i] == n) {
37             return false;
38         }
39     }
40     return true;
41 }
42
43 void sort(int v, edge&g[10][500]) {
44     int y = 0, r = 0;
45
46     while (g[v][r].v1 != -858993460) {
47         y++;
48         r++;
49     }
50     for (int i = 0; i < y; i++) {
51         for (int j = 0; j < y; j++) {
52             if (g[v][i].w < g[v][j].w) {
53                 edge temp = g[v][i];
54                 g[v][i] = g[v][j];
55                 g[v][j] = temp;
56             }
57         }
58     }
59 }
60
61 int main()
62 {
63     edge g[500];
64     edge numOfE[500];
65     edge structures[10][500];
66     int nV;
67     int numOfV[100];
68
69     f = fopen("f.txt", "r");
70     int s = 1, n = 0;
71     char str[255];
72     while (fgets(str, 255, f))
73     {
74         edge edgeWeight;
75         char* str1 = strtok(str, " ");
76         int w, i = 1;
77         while (str1 != 0) {
78             w = stoi(str1);
79             edgeWeight.v1 = s;
80             edgeWeight.v2 = i;
81             edgeWeight.w = w;
82             str1 = strtok(NULL, " ");
83             i++;
84             g[n] = edgeWeight;
85             n++;
86         }
87         s++;
88     }
89
90     int min = findMin(g, n);
91     int max = findMax(g, n);
92     int k = 0, k1 = 0;
93     int s1 = 0, s2 = 0, p = 0;
94     while (min <= max) {
95         for (int i = 0; i < n; i++) {
96             if (g[i].w == min && g[i].w != 0) {
97                 bool b1 = check(numOfV, g[i].v1);
98                 bool b2 = check(numOfV, g[i].v2);
99                 if (b1 && b2 && s1 == 0) {
100
101                     numOfV[k] = g[i].v1;
102                     numOfE[k1] = g[i];
103                     k++;
104                     k1++;
105                     numOfV[k] = g[i].v2;
106                     k++;
107                     k1++;
108                     if (p > 0) { s1++; s2 = 0; };
109                     structures[s1][s2] = g[i];
110                     p++;
111                     s2++;
112                 }
113                 else if (b1 && b2 && s1 > 0) {
114                     numOfV[k] = g[i].v1;
115                     numOfE[k1] = g[i];
116                     k++;
117                     k1++;
118                     numOfV[k] = g[i].v2;
119                     k++;
120                     k1++;
121                     s1++;
122                     s2 = 0;
123                     structures[s1][s2] = g[i];
124                     s2++;
125                 }
126                 else if (!b1 && !b2 && s1 > 0) {
127                     int v1 = 0, v2 = 0, sleep = 0;
128                     for (int q = 0; q < 10; q++) {
129                         for (int j = 0; j < 100; j++) {
130                             if (structures[q][j].v1 == g[i].v1) {
131                                 v1 = q;
132                             }
133                             else if (structures[q][j].v2 == g[i].v2) {
134                                 v2 = q;
135                             }
136                         }
137                     }
138                     if (v1 > v2) {
139                         int tem = v1;
140                         v1 = v2;
141                         v2 = tem;
142                     }
143
144                     if (v1 != v2) {
145                         numOfE[k1] = g[i];
146                         k1++;
147
148                         edge temp2[100];
149                         for (int i = 0; i < 100; i++) {
150                             temp2[i] = structures[v1][i];
151                         }
152
153                         int m = 0;
154                         for (int j = 0; j < 100; j++) {
155                             if (structures[v1][j].v1 == -858993460) {
156                                 structures[v1][j] = structures[v2][m];
157                                 m++;
158                             }
159                         }
160                         int y = 0;
161                         while (structures[v1][y].v1 != -858993460) {
162                             y++;
163                         }
164                         structures[v1][y] = g[i];
165
166                     for (int j = 0; j < 100; j++) {
167                         structures[v2][j].v1 = 0;
168                         structures[v2][j].v2 = 0;
169                         structures[v2][j].w = 0;
170                         sleep++;
171                     }
172                     sort(v2, structures);
173                     if (sleep < 0) {
174                         s1--;
175                     }
176                     else if (b2) {
177                         numOfV[k] = g[i].v1;
178                         numOfE[k1] = g[i];
179                         k++;
180                         k1++;
181                     }
182                     int w;
183                     for (int i = 0; i < 10; i++) {
184                         for (int j = 0; j < 100; j++) {
185                             if (structures[i][j].v2 == g[i].v2 || structures[i][j].v1 == g[i].v2) {
186                                 w = i;
187                                 break;
188                             }
189                         }
190                     }
191                     int y = 0;
192                     while (structures[w][y].v1 != -858993460) {
193                         y++;
194                     }
195                     structures[w][y] = g[i];
196                     sort(w, structures);
197                 }
198                 else if (b2) {
199                     numOfV[k] = g[i].v2;
200                     numOfE[k1] = g[i];
201
202                     k++;
203                     k1++;
204                     int w;
205                     for (int i = 0; i < 10; i++) {
206                         for (int j = 0; j < 100; j++) {
207                             if (structures[i][j].v2 == g[i].v1 || structures[i][j].v1 == g[i].v1) {
208                                 w = i;
209                                 break;
210                             }
211                         }
212                     }
213                     int y = 0;
214                     while (structures[w][y].v1 != -858993460) {
215                         y++;
216                     }
217                     structures[w][y] = g[i];
218                     sort(w, structures);
219                 }
220             }
221             min++;
222
223             cout << "nodes: (" << endl;
224             for (int i = 0; i < numV(n); i++) {
225                 cout << numOfV[i] << " ";
226             }
227             cout << ")" << endl;
228
229             cout << "edges: (" << endl;
230             for (int i = 0; i < k1; i++) {
231                 cout << "(" << numOfE[i].v1 << " ";
232                 cout << numOfE[i].v2 << " ";
233                 cout << ")" << endl;
234             }
235         }
236     }
237 }

```

Результат виконаної програми:


```

Консоль отладки Microsoft Visual Studio
vertices: {1, 4, 3, 5, 2, 7, 6, 10, 9, 11, 8, }
edges: {(1; 4), (3; 5), (2; 5), (2; 7), (4; 6), (3; 6), (6; 10), (7; 9), (9; 11), (8; 11), }

C:\Users\Admin\OneDrive\University\1 семестр\Дискретна Математика\Лаби\Л4\VY_DM_LR_4\Debug\VY_DM_LR_4.exe (процесс 6424)
завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрывать консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

Завдання № 6

Розв'язати задачу комівояжера для повного 8-ми вершинного 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	∞

Вершина	Маршрут	Вага
-	1	0
8	1-8	1
5	1-8-5	3
4	1-8-5-4	4
7	1-8-5-4-7	5
2	1-8-5-4-7-2	9
6	1-8-5-4-7-2-6	10
3	1-8-5-4-7-2-6-3	16
1	1-8-5-4-7-2-6-3-1	21

Програмна реалізація:

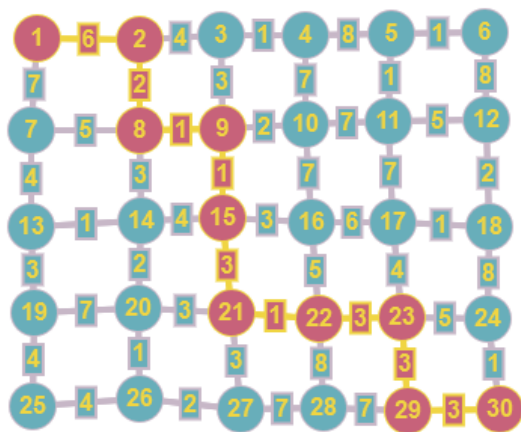
```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6
7 bool check(int key, int* mas, int kol) {
8     for (int j = 0; j < kol; j++)
9         if (mas[j] == key)
10             return false;
11     return true;
12 }
13
14 int main() {
15     int kol;
16     do
17     {
18         cout << "Enter the number of cities(2-10) --> ";
19         cin >> kol;
20     } while (kol < 2 || kol > 10);
21     int** arr = new int* [kol];
22     for (int i = 0; i < kol; i++)
23         arr[i] = new int[kol];
24
25     int rasst;
26     for (int i = 0; i < kol; i++) {
27         for (int j = 1; j < kol; j++) {
28             if (i == j)
29             {
30                 arr[i][j] = 0;
31                 continue;
32             }
33             do
34             {
35                 cout << "Enter the distance from the city " << i << " to the city " << j << " --> ";
36                 cin >> rasst;
37             } while (rasst < 1);
38             arr[i][j] = arr[j][i] = rasst;
39         }
40     }
41     cout << endl << "Adjacency matrix : ";
42     for (int i = 0; i < kol; i++) {
43         cout << endl;
44         for (int j = 0; j < kol; j++)
45             cout << setw(5) << arr[i][j];
46     }
47
48     int* route = new int[kol];
49
50     cout << endl;
51     char ans;
52     int start;
53     do {
54         for (int i = 0; i < kol; i++)
55             route[i] = -1;
56         do
57         {
58             cout << "Enter your starting city--> ";
59             cin >> start;
60         } while (start < 0 || start > kol - 1);
61
62         route[0] = start;
63         int now = start;
64         int path = 0;
65         cout << "\nRoute:" << endl;
66         for (int i = 1; i < kol; i++) {
67             int min = INT_MAX, min_town;
68             for (int j = 0; j < kol; j++) {
69                 if (check(j, route, kol) && arr[now][j] < min && arr[now][j] > 0) {
70                     min = arr[now][j];
71                     min_town = j;
72                 }
73             }
74
75             path += min;
76             route[i] = min_town;
77             cout << setw(2) << now << " -> " << setw(2) << route[i] << " (distance " << min << ", way " << path << ")" << endl;
78             now = route[i];
79         }
80         path += arr[start][now];
81         cout << setw(2) << now << " -> " << setw(2) << start << " (distance " << arr[start][now] << ", way " << path << ")" << endl;
82         cout << "Total distance traveled: " << path << endl;
83
84         cout << endl << "Would you like to continue your search for paths? (+, If yes) --> ";
85         cin >> ans;
86     } while (ans == '+');
87
88     delete[] route;
89     for (int i = 0; i < kol; i++)
90         delete[] arr[i];
91     delete[] arr;
92
93     return 0;
94 }
95 }
```

Результат виконаної програми:

```
Enter the number of cities(2-10) --> 8
Adjacency matrix :
Enter the distance from the city 0 to the city 1 --> 5 0 5 5 3 3 4 4 1
Enter the distance from the city 0 to the city 2 --> 5 5 0 4 3 2 1 4 6
Enter the distance from the city 0 to the city 3 --> 3 5 4 0 4 5 6 5 5
Enter the distance from the city 0 to the city 5 --> 4 3 3 4 0 1 5 1 7
Enter the distance from the city 0 to the city 6 --> 4 3 2 5 1 0 5 5 2
Enter the distance from the city 0 to the city 7 --> 1 4 1 6 5 5 0 7 3
Enter the distance from the city 1 to the city 2 --> 4 4 4 5 1 5 7 0 2
Enter the distance from the city 1 to the city 3 --> 3 1 6 5 7 2 3 2 0
Enter the distance from the city 1 to the city 4 --> 2
Enter the distance from the city 1 to the city 5 --> 1
Enter the distance from the city 1 to the city 6 --> 4
Enter the distance from the city 1 to the city 7 --> 6
Enter the distance from the city 2 to the city 3 --> 4
Enter the distance from the city 2 to the city 4 --> 5
Enter the distance from the city 2 to the city 5 --> 6
Enter the distance from the city 2 to the city 6 --> 5
Enter the distance from the city 2 to the city 7 --> 5
Enter the distance from the city 3 to the city 4 --> 1
Enter the distance from the city 3 to the city 5 --> 5
Enter the distance from the city 3 to the city 6 --> 1
Enter the distance from the city 3 to the city 7 --> 7
Enter the distance from the city 4 to the city 5 --> 5
Enter the distance from the city 4 to the city 6 --> 5
Enter the distance from the city 4 to the city 7 --> 2
Enter the distance from the city 5 to the city 6 --> 7
Enter the distance from the city 5 to the city 7 --> 3
Enter the distance from the city 6 to the city 7 --> 2
Route:
0 -> 7 (distance 1, way 1)
7 -> 4 (distance 2, way 3)
4 -> 3 (distance 1, way 4)
3 -> 6 (distance 1, way 5)
6 -> 1 (distance 4, way 9)
1 -> 5 (distance 1, way 10)
5 -> 2 (distance 6, way 16)
2 -> 0 (distance 5, way 21)
Total distance traveled: 21
```

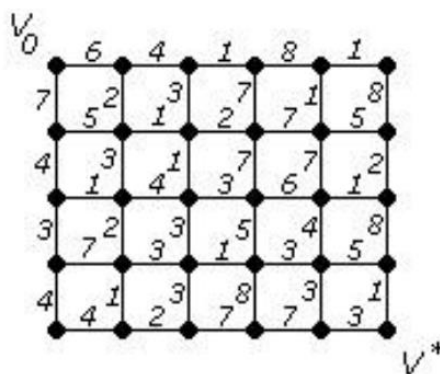
Завдання № 7

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



9)

]



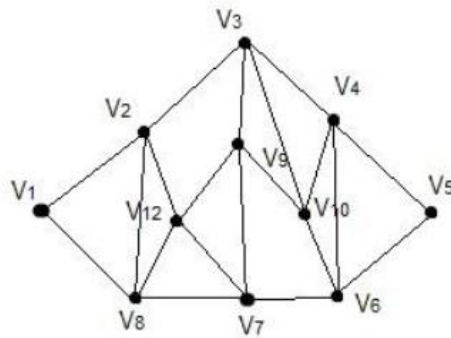
Програмна реалізація:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int n,m;
8     cout<<"Skilky verxshyn hochete?";
9     cin>>n;
10    cout<<"Skilky reber hochete? ";
11    cin>>m;
12    cout<<endl;
13
14    int s=n;
15    int a[s][s];
16    for (int i=0; i<n; i++)
17    {
18        for (int j=0; j<n; j++)
19        {
20            a[i][j]=0;
21        }
22    }
23
24    //введення даних
25    int t1,t2,w;
26    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
27    for (int k=0; k<m; k++)
28    {
29        cout<<"Rebro"<<k+1<<" : ";
30        cin>>t1>>t2>>w;
31        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
32        {
33            cout<<"The data entered incorrectly."<<endl;
34            cout<<"Try again please."<<endl;
35            cin>>t1>>t2>>w;
36        }
37        a[t1-1][t2-1]=w;
38        a[t2-1][t1-1]=w;
39    }
40    cout<<endl;
41
```

```
42    int sh[n]; //масив ширини
43    int v[n]; //масив висот
44    int temp, index, minn;
45    for (int i = 0; i<n; i++)
46    {
47        sh[i] = 300000;
48        v[i] = 1;
49    }
50    sh[0] = 0;
51
52    do
53    {
54        index = 300000;
55        minn = 300000;
56        for (int i = 0; i<n; i++)
57        {
58            //пошук мінімального значення
59            //якщо висота не менша за 300000
60            if ((v[i] == 1) && (sh[i]<minn))
61            {
62                //новий мінімум
63                minn = sh[i];
64                index = i;
65            }
66        }
67
68        //введення висоти
69        if (index != 300000)
70        {
71            for (int i = 0; i<n; i++)
72            {
73                if (a[index][i] > 0)
74                {
75                    temp = minn + a[index][i];
76                    if (temp < sh[i])
77                    {
78                        sh[i] = temp;
79                    }
80                }
81            }
82        }
83    } while (index != 300000);
84
85    //виведення результату
86    cout<<"Skilky verxshyn hochete?";
87    cin>>n;
88    cout<<"Skilky reber hochete? ";
89    cin>>m;
90    cout<<endl;
91
92    int s=n;
93    int a[s][s];
94    for (int i=0; i<n; i++)
95    {
96        for (int j=0; j<n; j++)
97        {
98            a[i][j]=0;
99        }
100    }
101
102    //введення даних
103    int t1,t2,w;
104    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
105    for (int k=0; k<m; k++)
106    {
107        cout<<"Rebro"<<k+1<<" : ";
108        cin>>t1>>t2>>w;
109        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
110        {
111            cout<<"The data entered incorrectly."<<endl;
112            cout<<"Try again please."<<endl;
113            cin>>t1>>t2>>w;
114        }
115        a[t1-1][t2-1]=w;
116        a[t2-1][t1-1]=w;
117    }
118    cout<<endl;
119
120    int sh[n]; //масив ширини
121    int v[n]; //масив висот
122    int temp, index, minn;
123    for (int i = 0; i<n; i++)
124    {
125        sh[i] = 300000;
126        v[i] = 1;
127    }
128    sh[0] = 0;
129
130    do
131    {
132        index = 300000;
133        minn = 300000;
134        for (int i = 0; i<n; i++)
135        {
136            //пошук мінімального значення
137            //якщо висота не менша за 300000
138            if ((v[i] == 1) && (sh[i]<minn))
139            {
140                //новий мінімум
141                minn = sh[i];
142                index = i;
143            }
144        }
145
146        //введення висоти
147        if (index != 300000)
148        {
149            for (int i = 0; i<n; i++)
150            {
151                if (a[index][i] > 0)
152                {
153                    temp = minn + a[index][i];
154                    if (temp < sh[i])
155                    {
156                        sh[i] = temp;
157                    }
158                }
159            }
160        }
161    } while (index != 300000);
162
163    //виведення результату
164    cout<<"Skilky verxshyn hochete?";
165    cin>>n;
166    cout<<"Skilky reber hochete? ";
167    cin>>m;
168    cout<<endl;
169
170    int s=n;
171    int a[s][s];
172    for (int i=0; i<n; i++)
173    {
174        for (int j=0; j<n; j++)
175        {
176            a[i][j]=0;
177        }
178    }
179
180    //введення даних
181    int t1,t2,w;
182    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
183    for (int k=0; k<m; k++)
184    {
185        cout<<"Rebro"<<k+1<<" : ";
186        cin>>t1>>t2>>w;
187        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
188        {
189            cout<<"The data entered incorrectly."<<endl;
190            cout<<"Try again please."<<endl;
191            cin>>t1>>t2>>w;
192        }
193        a[t1-1][t2-1]=w;
194        a[t2-1][t1-1]=w;
195    }
196    cout<<endl;
197
198    int sh[n]; //масив ширини
199    int v[n]; //масив висот
200    int temp, index, minn;
201    for (int i = 0; i<n; i++)
202    {
203        sh[i] = 300000;
204        v[i] = 1;
205    }
206    sh[0] = 0;
207
208    do
209    {
210        index = 300000;
211        minn = 300000;
212        for (int i = 0; i<n; i++)
213        {
214            //пошук мінімального значення
215            //якщо висота не менша за 300000
216            if ((v[i] == 1) && (sh[i]<minn))
217            {
218                //новий мінімум
219                minn = sh[i];
220                index = i;
221            }
222        }
223
224        //введення висоти
225        if (index != 300000)
226        {
227            for (int i = 0; i<n; i++)
228            {
229                if (a[index][i] > 0)
230                {
231                    temp = minn + a[index][i];
232                    if (temp < sh[i])
233                    {
234                        sh[i] = temp;
235                    }
236                }
237            }
238        }
239    } while (index != 300000);
240
241    //виведення результату
242    cout<<"Skilky verxshyn hochete?";
243    cin>>n;
244    cout<<"Skilky reber hochete? ";
245    cin>>m;
246    cout<<endl;
247
248    int s=n;
249    int a[s][s];
250    for (int i=0; i<n; i++)
251    {
252        for (int j=0; j<n; j++)
253        {
254            a[i][j]=0;
255        }
256    }
257
258    //введення даних
259    int t1,t2,w;
260    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
261    for (int k=0; k<m; k++)
262    {
263        cout<<"Rebro"<<k+1<<" : ";
264        cin>>t1>>t2>>w;
265        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
266        {
267            cout<<"The data entered incorrectly."<<endl;
268            cout<<"Try again please."<<endl;
269            cin>>t1>>t2>>w;
270        }
271        a[t1-1][t2-1]=w;
272        a[t2-1][t1-1]=w;
273    }
274    cout<<endl;
275
276    int sh[n]; //масив ширини
277    int v[n]; //масив висот
278    int temp, index, minn;
279    for (int i = 0; i<n; i++)
280    {
281        sh[i] = 300000;
282        v[i] = 1;
283    }
284    sh[0] = 0;
285
286    do
287    {
288        index = 300000;
289        minn = 300000;
290        for (int i = 0; i<n; i++)
291        {
292            //пошук мінімального значення
293            //якщо висота не менша за 300000
294            if ((v[i] == 1) && (sh[i]<minn))
295            {
296                //новий мінімум
297                minn = sh[i];
298                index = i;
299            }
300        }
301
302        //введення висоти
303        if (index != 300000)
304        {
305            for (int i = 0; i<n; i++)
306            {
307                if (a[index][i] > 0)
308                {
309                    temp = minn + a[index][i];
310                    if (temp < sh[i])
311                    {
312                        sh[i] = temp;
313                    }
314                }
315            }
316        }
317    } while (index != 300000);
318
319    //виведення результату
320    cout<<"Skilky verxshyn hochete?";
321    cin>>n;
322    cout<<"Skilky reber hochete? ";
323    cin>>m;
324    cout<<endl;
325
326    int s=n;
327    int a[s][s];
328    for (int i=0; i<n; i++)
329    {
330        for (int j=0; j<n; j++)
331        {
332            a[i][j]=0;
333        }
334    }
335
336    //введення даних
337    int t1,t2,w;
338    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
339    for (int k=0; k<m; k++)
340    {
341        cout<<"Rebro"<<k+1<<" : ";
342        cin>>t1>>t2>>w;
343        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
344        {
345            cout<<"The data entered incorrectly."<<endl;
346            cout<<"Try again please."<<endl;
347            cin>>t1>>t2>>w;
348        }
349        a[t1-1][t2-1]=w;
350        a[t2-1][t1-1]=w;
351    }
352    cout<<endl;
353
354    int sh[n]; //масив ширини
355    int v[n]; //масив висот
356    int temp, index, minn;
357    for (int i = 0; i<n; i++)
358    {
359        sh[i] = 300000;
360        v[i] = 1;
361    }
362    sh[0] = 0;
363
364    do
365    {
366        index = 300000;
367        minn = 300000;
368        for (int i = 0; i<n; i++)
369        {
370            //пошук мінімального значення
371            //якщо висота не менша за 300000
372            if ((v[i] == 1) && (sh[i]<minn))
373            {
374                //новий мінімум
375                minn = sh[i];
376                index = i;
377            }
378        }
379
380        //введення висоти
381        if (index != 300000)
382        {
383            for (int i = 0; i<n; i++)
384            {
385                if (a[index][i] > 0)
386                {
387                    temp = minn + a[index][i];
388                    if (temp < sh[i])
389                    {
390                        sh[i] = temp;
391                    }
392                }
393            }
394        }
395    } while (index != 300000);
396
397    //виведення результату
398    cout<<"Skilky verxshyn hochete?";
399    cin>>n;
400    cout<<"Skilky reber hochete? ";
401    cin>>m;
402    cout<<endl;
403
404    int s=n;
405    int a[s][s];
406    for (int i=0; i<n; i++)
407    {
408        for (int j=0; j<n; j++)
409        {
410            a[i][j]=0;
411        }
412    }
413
414    //введення даних
415    int t1,t2,w;
416    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
417    for (int k=0; k<m; k++)
418    {
419        cout<<"Rebro"<<k+1<<" : ";
420        cin>>t1>>t2>>w;
421        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
422        {
423            cout<<"The data entered incorrectly."<<endl;
424            cout<<"Try again please."<<endl;
425            cin>>t1>>t2>>w;
426        }
427        a[t1-1][t2-1]=w;
428        a[t2-1][t1-1]=w;
429    }
430    cout<<endl;
431
432    int sh[n]; //масив ширини
433    int v[n]; //масив висот
434    int temp, index, minn;
435    for (int i = 0; i<n; i++)
436    {
437        sh[i] = 300000;
438        v[i] = 1;
439    }
440    sh[0] = 0;
441
442    do
443    {
444        index = 300000;
445        minn = 300000;
446        for (int i = 0; i<n; i++)
447        {
448            //пошук мінімального значення
449            //якщо висота не менша за 300000
450            if ((v[i] == 1) && (sh[i]<minn))
451            {
452                //новий мінімум
453                minn = sh[i];
454                index = i;
455            }
456        }
457
458        //введення висоти
459        if (index != 300000)
460        {
461            for (int i = 0; i<n; i++)
462            {
463                if (a[index][i] > 0)
464                {
465                    temp = minn + a[index][i];
466                    if (temp < sh[i])
467                    {
468                        sh[i] = temp;
469                    }
470                }
471            }
472        }
473    } while (index != 300000);
474
475    //виведення результату
476    cout<<"Skilky verxshyn hochete?";
477    cin>>n;
478    cout<<"Skilky reber hochete? ";
479    cin>>m;
480    cout<<endl;
481
482    int s=n;
483    int a[s][s];
484    for (int i=0; i<n; i++)
485    {
486        for (int j=0; j<n; j++)
487        {
488            a[i][j]=0;
489        }
490    }
491
492    //введення даних
493    int t1,t2,w;
494    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
495    for (int k=0; k<m; k++)
496    {
497        cout<<"Rebro"<<k+1<<" : ";
498        cin>>t1>>t2>>w;
499        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
500        {
501            cout<<"The data entered incorrectly."<<endl;
502            cout<<"Try again please."<<endl;
503            cin>>t1>>t2>>w;
504        }
505        a[t1-1][t2-1]=w;
506        a[t2-1][t1-1]=w;
507    }
508    cout<<endl;
509
510    int sh[n]; //масив ширини
511    int v[n]; //масив висот
512    int temp, index, minn;
513    for (int i = 0; i<n; i++)
514    {
515        sh[i] = 300000;
516        v[i] = 1;
517    }
518    sh[0] = 0;
519
520    do
521    {
522        index = 300000;
523        minn = 300000;
524        for (int i = 0; i<n; i++)
525        {
526            //пошук мінімального значення
527            //якщо висота не менша за 300000
528            if ((v[i] == 1) && (sh[i]<minn))
529            {
530                //новий мінімум
531                minn = sh[i];
532                index = i;
533            }
534        }
535
536        //введення висоти
537        if (index != 300000)
538        {
539            for (int i = 0; i<n; i++)
540            {
541                if (a[index][i] > 0)
542                {
543                    temp = minn + a[index][i];
544                    if (temp < sh[i])
545                    {
546                        sh[i] = temp;
547                    }
548                }
549            }
550        }
551    } while (index != 300000);
552
553    //виведення результату
554    cout<<"Skilky verxshyn hochete?";
555    cin>>n;
556    cout<<"Skilky reber hochete? ";
557    cin>>m;
558    cout<<endl;
559
560    int s=n;
561    int a[s][s];
562    for (int i=0; i<n; i++)
563    {
564        for (int j=0; j<n; j++)
565        {
566            a[i][j]=0;
567        }
568    }
569
570    //введення даних
571    int t1,t2,w;
572    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
573    for (int k=0; k<m; k++)
574    {
575        cout<<"Rebro"<<k+1<<" : ";
576        cin>>t1>>t2>>w;
577        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
578        {
579            cout<<"The data entered incorrectly."<<endl;
580            cout<<"Try again please."<<endl;
581            cin>>t1>>t2>>w;
582        }
583        a[t1-1][t2-1]=w;
584        a[t2-1][t1-1]=w;
585    }
586    cout<<endl;
587
588    int sh[n]; //масив ширини
589    int v[n]; //масив висот
590    int temp, index, minn;
591    for (int i = 0; i<n; i++)
592    {
593        sh[i] = 300000;
594        v[i] = 1;
595    }
596    sh[0] = 0;
597
598    do
599    {
600        index = 300000;
601        minn = 300000;
602        for (int i = 0; i<n; i++)
603        {
604            //пошук мінімального значення
605            //якщо висота не менша за 300000
606            if ((v[i] == 1) && (sh[i]<minn))
607            {
608                //новий мінімум
609                minn = sh[i];
610                index = i;
611            }
612        }
613
614        //введення висоти
615        if (index != 300000)
616        {
617            for (int i = 0; i<n; i++)
618            {
619                if (a[index][i] > 0)
620                {
621                    temp = minn + a[index][i];
622                    if (temp < sh[i])
623                    {
624                        sh[i] = temp;
625                    }
626                }
627            }
628        }
629    } while (index != 300000);
630
631    //виведення результату
632    cout<<"Skilky verxshyn hochete?";
633    cin>>n;
634    cout<<"Skilky reber hochete? ";
635    cin>>m;
636    cout<<endl;
637
638    int s=n;
639    int a[s][s];
640    for (int i=0; i<n; i++)
641    {
642        for (int j=0; j<n; j++)
643        {
644            a[i][j]=0;
645        }
646    }
647
648    //введення даних
649    int t1,t2,w;
650    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
651    for (int k=0; k<m; k++)
652    {
653        cout<<"Rebro"<<k+1<<" : ";
654        cin>>t1>>t2>>w;
655        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
656        {
657            cout<<"The data entered incorrectly."<<endl;
658            cout<<"Try again please."<<endl;
659            cin>>t1>>t2>>w;
660        }
661        a[t1-1][t2-1]=w;
662        a[t2-1][t1-1]=w;
663    }
664    cout<<endl;
665
666    int sh[n]; //масив ширини
667    int v[n]; //масив висот
668    int temp, index, minn;
669    for (int i = 0; i<n; i++)
670    {
671        sh[i] = 300000;
672        v[i] = 1;
673    }
674    sh[0] = 0;
675
676    do
677    {
678        index = 300000;
679        minn = 300000;
680        for (int i = 0; i<n; i++)
681        {
682            //пошук мінімального значення
683            //якщо висота не менша за 300000
684            if ((v[i] == 1) && (sh[i]<minn))
685            {
686                //новий мінімум
687                minn = sh[i];
688                index = i;
689            }
690        }
691
692        //введення висоти
693        if (index != 300000)
694        {
695            for (int i = 0; i<n; i++)
696            {
697                if (a[index][i] > 0)
698                {
699                    temp = minn + a[index][i];
700                    if (temp < sh[i])
701                    {
702                        sh[i] = temp;
703                    }
704                }
705            }
706        }
707    } while (index != 300000);
708
709    //виведення результату
710    cout<<"Skilky verxshyn hochete?";
711    cin>>n;
712    cout<<"Skilky reber hochete? ";
713    cin>>m;
714    cout<<endl;
715
716    int s=n;
717    int a[s][s];
718    for (int i=0; i<n; i++)
719    {
720        for (int j=0; j<n; j++)
721        {
722            a[i][j]=0;
723        }
724    }
725
726    //введення даних
727    int t1,t2,w;
728    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
729    for (int k=0; k<m; k++)
730    {
731        cout<<"Rebro"<<k+1<<" : ";
732        cin>>t1>>t2>>w;
733        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
734        {
735            cout<<"The data entered incorrectly."<<endl;
736            cout<<"Try again please."<<endl;
737            cin>>t1>>t2>>w;
738        }
739        a[t1-1][t2-1]=w;
740        a[t2-1][t1-1]=w;
741    }
742    cout<<endl;
743
744    int sh[n]; //масив ширини
745    int v[n]; //масив висот
746    int temp, index, minn;
747    for (int i = 0; i<n; i++)
748    {
749        sh[i] = 300000;
750        v[i] = 1;
751    }
752    sh[0] = 0;
753
754    do
755    {
756        index = 300000;
757        minn = 300000;
758        for (int i = 0; i<n; i++)
759        {
760            //пошук мінімального значення
761            //якщо висота не менша за 300000
762            if ((v[i] == 1) && (sh[i]<minn))
763            {
764                //новий мінімум
765                minn = sh[i];
766                index = i;
767            }
768        }
769
770        //введення висоти
771        if (index != 300000)
772        {
773            for (int i = 0; i<n; i++)
774            {
775                if (a[index][i] > 0)
776                {
777                    temp = minn + a[index][i];
778                    if (temp < sh[i])
779                    {
780                        sh[i] = temp;
781                    }
782                }
783            }
784        }
785    } while (index != 300000);
786
787    //виведення результату
788    cout<<"Skilky verxshyn hochete?";
789    cin>>n;
790    cout<<"Skilky reber hochete? ";
791    cin>>m;
792    cout<<endl;
793
794    int s=n;
795    int a[s][s];
796    for (int i=0; i<n; i++)
797    {
798        for (int j=0; j<n; j++)
799        {
800            a[i][j]=0;
801        }
802    }
803
804    //введення даних
805    int t1,t2,w;
806    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
807    for (int k=0; k<m; k++)
808    {
809        cout<<"Rebro"<<k+1<<" : ";
810        cin>>t1>>t2>>w;
811        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
812        {
813            cout<<"The data entered incorrectly."<<endl;
814            cout<<"Try again please."<<endl;
815            cin>>t1>>t2>>w;
816        }
817        a[t1-1][t2-1]=w;
818        a[t2-1][t1-1]=w;
819    }
820    cout<<endl;
821
822    int sh[n]; //масив ширини
823    int v[n]; //масив висот
824    int temp, index, minn;
825    for (int i = 0; i<n; i++)
826    {
827        sh[i] = 300000;
828        v[i] = 1;
829    }
830    sh[0] = 0;
831
832    do
833    {
834        index = 300000;
835        minn = 300000;
836        for (int i = 0; i<n; i++)
837        {
838            //пошук мінімального значення
839            //якщо висота не менша за 300000
840            if ((v[i] == 1) && (sh[i]<minn))
841            {
842                //новий мінімум
843                minn = sh[i];
844                index = i;
845            }
846        }
847
848        //введення висоти
849        if (index != 300000)
850        {
851            for (int i = 0; i<n; i++)
852            {
853                if (a[index][i] > 0)
854                {
855                    temp = minn + a[index][i];
856                    if (temp < sh[i])
857                    {
858                        sh[i] = temp;
859                    }
860                }
861            }
862        }
863    } while (index != 300000);
864
865    //виведення результату
866    cout<<"Skilky verxshyn hochete?";
867    cin>>n;
868    cout<<"Skilky reber hochete? ";
869    cin>>m;
870    cout<<endl;
871
872    int s=n;
873    int a[s][s];
874    for (int i=0; i<n; i++)
875    {
876        for (int j=0; j<n; j++)
877        {
878            a[i][j]=0;
879        }
880    }
881
882    //введення даних
883    int t1,t2,w;
884    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
885    for (int k=0; k<m; k++)
886    {
887        cout<<"Rebro"<<k+1<<" : ";
888        cin>>t1>>t2>>w;
889        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
890        {
891            cout<<"The data entered incorrectly."<<endl;
892            cout<<"Try again please."<<endl;
893            cin>>t1>>t2>>w;
894        }
895        a[t1-1][t2-1]=w;
896        a[t2-1][t1-1]=w;
897    }
898    cout<<endl;
899
900    int sh[n]; //масив ширини
901    int v[n]; //масив висот
902    int temp, index, minn;
903    for (int i = 0; i<n; i++)
904    {
905        sh[i] = 300000;
906        v[i] = 1;
907    }
908    sh[0] = 0;
909
910    do
911    {
912        index = 300000;
913        minn = 300000;
914        for (int i = 0; i<n; i++)
915        {
916            //пошук мінімального значення
917            //якщо висота не менша за 300000
918            if ((v[i] == 1) && (sh[i]<minn))
919            {
920                //новий мінімум
921                minn = sh[i];
922                index = i;
923            }
924        }
925
926        //введення висоти
927        if (index != 300000)
928        {
929            for (int i = 0; i<n; i++)
930            {
931                if (a[index][i] > 0)
932                {
933                    temp = minn + a[index][i];
934                    if (temp < sh[i])
935                    {
936                        sh[i] = temp;
937                    }
938                }
939            }
940        }
941    } while (index != 300000);
942
943    //виведення результату
944    cout<<"Skilky verxshyn hochete?";
945    cin>>n;
946    cout<<"Skilky reber hochete? ";
947    cin>>m;
948    cout<<endl;
949
950    int s=n;
951    int a[s][s];
952    for (int i=0; i<n; i++)
953    {
954        for (int j=0; j<n; j++)
955        {
956            a[i][j]=0;
957        }
958    }
959
960    //введення даних
961    int t1,t2,w;
962    cout<<"Enter rebra(1 verxshyna | 2 verxshyna | waga):\n";
963    for (int k=0; k<m; k++)
964    {
965        cout<<"Rebro"<<k+1<<" : ";
966        cin>>t1>>t2>>w;
967        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
968        {
969            cout<<"The data entered incorrectly."<<endl;
970            cout<<"Try again please."<<endl;
971            cin>>t1>>t2>>w;
972        }
973        a[t1-1][t2-1]=w;
974        a[t2-1][t1-1]=w;
975    }
976    cout<<endl;
977
978    int sh[n]; //масив ширини
979    int v
```

Завдання № 8

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



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

програмна реалізація

```
1  #include<iostream>
2  #define NODE 11
3  using namespace std;
4  int graph[NODE][NODE] = {
5      {0,1,0,0,0,0,0,1,0,0,0},
6      {1,0,1,0,0,0,0,1,0,0,1},
7      {0,1,0,1,0,0,0,0,1,1,0},
8      {0,0,1,0,1,1,0,0,0,1,0},
9      {0,0,0,1,0,1,0,0,0,0,0},
10     {0,0,0,1,0,1,0,0,0,1,0},
11     {0,0,0,0,0,1,0,1,1,0,1},
12     {1,1,0,0,0,0,1,0,0,0,1},
13     {0,0,1,0,0,0,1,0,0,1,1},
14     {0,0,1,1,0,0,1,0,0,1,0},
15     {0,1,0,0,0,0,1,1,1,0,0}
16 };
17 int tempGraph[NODE][NODE];
18 int findStartVert() {
19     for (int i = 1; i < NODE; i++) {
20         int deg = 0;
21         for (int j = 0; j < NODE; j++) {
22             if (tempGraph[i][j])
23                 deg++;
24         }
25         if (deg % 2 != 0)
26             return i;
27     }
28     return 0;
29 }
30 bool isBridge(int u, int v) {
31     int deg = 0;
32     for (int i = 0; i < NODE; i++)
33         if (tempGraph[v][i])
34             deg++;
35     if (deg > 1) {
36         return false;
37     }
38     return true;
39 }
```

Результат виконаної програми:

```
40 int edgeCount() {
41     int count = 0;
42     for (int i = 0; i < NODE; i++)
43         for (int j = 0; j < NODE; j++)
44             if (tempGraph[i][j])
45                 count++;
46     return count;
47 }
48 void fleuryAlgorithm(int start) {
49     static int edge = edgeCount();
50     for (int v = 0; v < NODE; v++) {
51         if (tempGraph[start][v]) {
52             if (edge <= 1 || !isBridge(start, v)) {
53                 cout << start + 1 << "--" << v + 1 << " " << endl;
54                 tempGraph[start][v] = tempGraph[v][start] = 0;
55                 edge--;
56                 fleuryAlgorithm(v);
57             }
58         }
59     }
60 }
61 int main() {
62     for (int i = 0; i < NODE; i++)
63         for (int j = 0; j < NODE; j++)
64             tempGraph[i][j] = graph[i][j];
65     cout << "Euler Path Or Circuit: \n";
66     fleuryAlgorithm(findStartVert());
67 }
```

Microsoft Visual Studio Debug

Euler Path Or Circuit:

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

6)

1) $1 - 2 - 8 - 1 +$

2) $2 - 3 - 9 - 12 - 2 +$

3) $12 - 8 - 7 - 12$

4) $7 - 9 - 10 - 6 - 7$

5) $3 - 10 - 4 - 3$

5) $6 - 4 - 5 - 6$

▪ $1 - 2 - 8 - 1$

▪ $1 - 2 - 3 - 9 - 12 - 2 - 8 - 1$

▪ $1 - 2 - 3 - 9 - 12 - 2 - 8 - 12 - 7 - 9 - 10 - 6 - 7 - 8 - 1$

▪ $1 - 2 - 3 - 9 - 12 - 2 - 8 - 12 - 7 - 9 - 10 - 6 - 4 - 10 - 3 - 4 - 5 - 6 - 7 - 8 - 1$

Завдання №9

Спростити формули (привести їх до скороченої ДНФ).

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

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

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

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

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

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

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

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

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

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

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