```
/*****************************************************************

                        Online C Compiler.
                Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile a

*****************************************************************/
#include <stdio.h>

#define MAX 30

typedef struct edge {
  int u, v, w;
} edge;

typedef struct edge_list {
  edge data[MAX];
  int n;
} edge_list;

edge_list elist;

int Graph[MAX][MAX], n;
edge_list spanlist;

void kruskalAlgo();
int find(int belongs[], int vertexno);
void applyUnion(int belongs[], int c1, int c2);
void sort();
void print();


void kruskalAlgo() {
  int belongs[MAX], i, j, cno1, cno2;
  elist.n = 0;

  for (i = 1; i < n; i++)
    for (j = 0; j < i; j++) {
```

input

```
ain.c
38      for (j = 0; j < i; j++) {
39        if (Graph[i][j] != 0) {
40          elist.data[elist.n].u = i;
41          elist.data[elist.n].v = j;
42          elist.data[elist.n].w = Graph[i][j];
43          elist.n++;
44        }
45      }
46
47    sort();
48
49    for (i = 0; i < n; i++)
50      belongs[i] = i;
51
52    spanlist.n = 0;
53
54    for (i = 0; i < elist.n; i++) {
55      cno1 = find(belongs, elist.data[i].u);
56      cno2 = find(belongs, elist.data[i].v);
57
58      if (cno1 != cno2) {
59        spanlist.data[spanlist.n] = elist.data[i];
60        spanlist.n = spanlist.n + 1;
61        applyUnion(belongs, cno1, cno2);
62      }
63    }
64  }
65
66  int find(int belongs[], int vertexno) {
67    return (belongs[vertexno]);
68  }
69
70  void applyUnion(int belongs[], int c1, int c2) {
71    int i;
72
73    for (i = 0; i < n; i++)
74      if (belongs[i] == c2)
75        belongs[i] = c1;
```

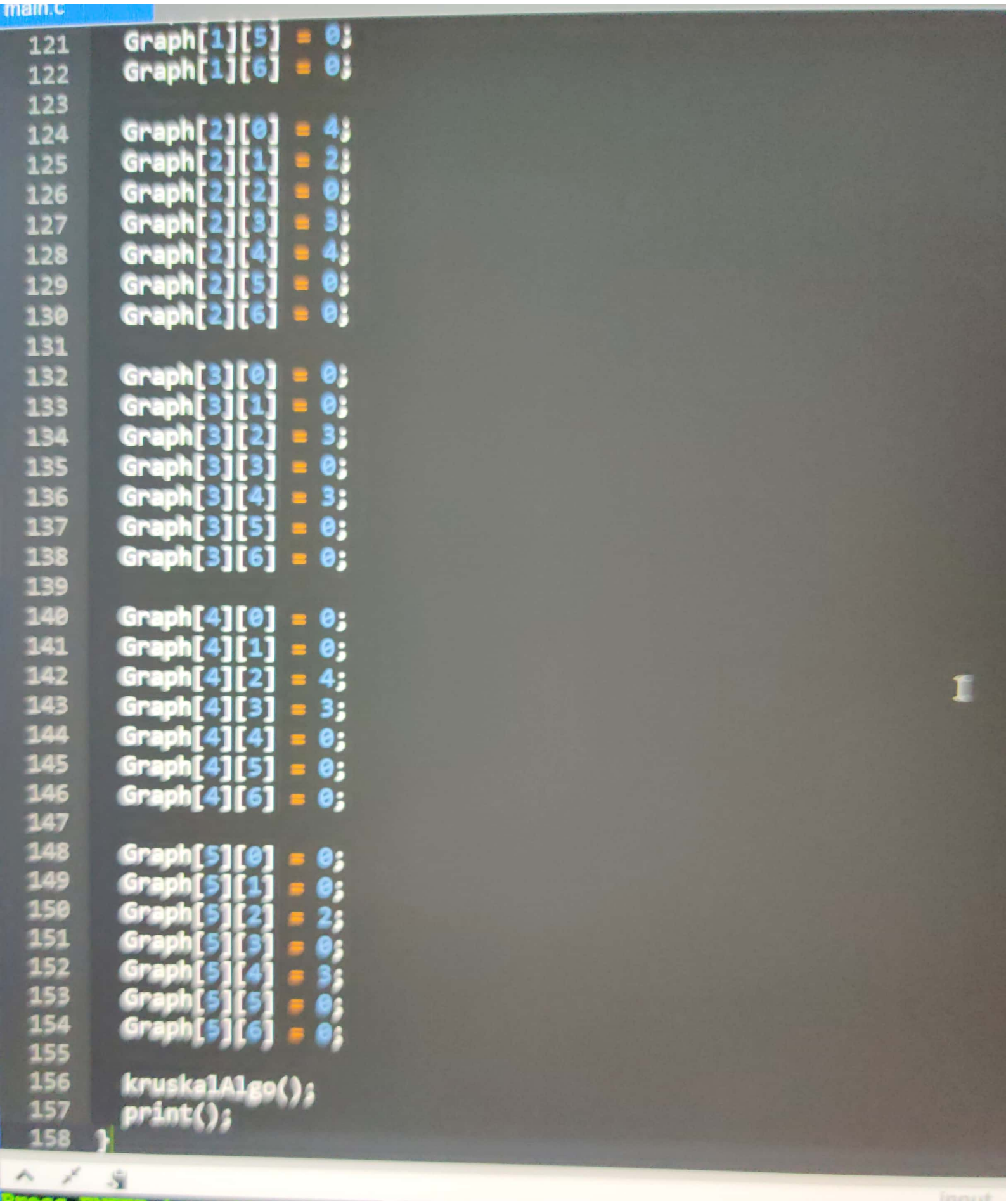
input

ain.c

```
74      if (belongs[i] == c2)
75        belongs[i] = c1;
76  }
77
78
79  void sort() {
80    int i, j;
81    edge temp;
82
83    for (i = 1; i < elist.n; i++)
84      for (j = 0; j < elist.n - 1; j++)
85        if (elist.data[j].w > elist.data[j + 1].w) {
86          temp = elist.data[j];
87          elist.data[j] = elist.data[j + 1];
88          elist.data[j + 1] = temp;
89        }
90  }
91
92
93  void print() {
94    int i, cost = 0;
95
96    for (i = 0; i < spanlist.n; i++) {
97      printf("\n%d - %d : %d", spanlist.data[i].u, spanlist.data[i].v, spanlist.data[i].w);
98      cost = cost + spanlist.data[i].w;
99    }
100
101   printf("\nSpanning tree cost: %d", cost);
102 }
103
104 int main() {
105
106   n = 6;
107
108   Graph[0][0] = 0;
109   Graph[0][1] = 4;
110   Graph[0][2] = 4;
111   Graph[0][3] = 0;
```

input

```
103
104 int main() {
105
106     n = 6;
107
108     Graph[0][0] = 0;
109     Graph[0][1] = 4;
110     Graph[0][2] = 4;
111     Graph[0][3] = 0;
112     Graph[0][4] = 0;
113     Graph[0][5] = 0;
114     Graph[0][6] = 0;
115
116     Graph[1][0] = 4;
117     Graph[1][1] = 0;
118     Graph[1][2] = 2;
119     Graph[1][3] = 0;
120     Graph[1][4] = 0;
121     Graph[1][5] = 0;
122     Graph[1][6] = 0;
123
124     Graph[2][0] = 4;
125     Graph[2][1] = 2;
126     Graph[2][2] = 0;
127     Graph[2][3] = 3;
128     Graph[2][4] = 4;
129     Graph[2][5] = 0;
130     Graph[2][6] = 0;
131
132     Graph[3][0] = 0;
133     Graph[3][1] = 0;
134     Graph[3][2] = 3;
135     Graph[3][3] = 0;
136     Graph[3][4] = 3;
137     Graph[3][5] = 0;
138     Graph[3][6] = 0;
139
140     Graph[4][0] = 0;
```



```
2 - 1 : 2
5 - 2 : 2
3 - 2 : 3
4 - 3 : 3
1 - 0 : 4
Spanning tree cost: 14

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