

```

#include <stdio.h>

#define INFINITY 9999

#define MAX 10

void Dijkstra(int Graph[MAX][MAX], int n, int start);

void Dijkstra(int Graph[MAX][MAX], int n, int start) {
    int cost[MAX][MAX], distance[MAX], pred[MAX];
    int visited[MAX], count, mindistance, nextnode, i, j;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            if (Graph[i][j] == 0)
                cost[i][j] = INFINITY;
            else
                cost[i][j] = Graph[i][j];

    for (i = 0; i < n; i++) {
        distance[i] = cost[start][i];
        pred[i] = start;
        visited[i] = 0;
    }

    distance[start] = 0;
    visited[start] = 1;
    count = 1;

    while (count < n - 1) {
        mindistance = INFINITY;

        for (i = 0; i < n; i++)
            if (distance[i] < mindistance && !visited[i]) {

```

```

        mindistance = distance[i];
        nextnode = i;
    }

    visited[nextnode] = 1;
    for (i = 0; i < n; i++)
        if (!visited[i])
            if (mindistance + cost[nextnode][i] < distance[i]) {
                distance[i] = mindistance + cost[nextnode][i];
                pred[i] = nextnode;
            }
    count++;
}
for (i = 0; i < n; i++)
    if (i != start) {
        printf("\nDistance from source to %d: %d", i, distance[i]);
    }
}

int main() {
    int Graph[MAX][MAX], i, j, n, u;
    n = 7;

    Graph[0][0] = 0;
    Graph[0][1] = 8;
    Graph[0][2] = 1;
    Graph[0][3] = 4;
    Graph[0][4] = 6;
    Graph[0][5] = 7;
    Graph[0][6] = 1;

    Graph[1][0] = 4;

```

Graph[1][1] = 0;  
Graph[1][2] = 2;  
Graph[1][3] = 0;  
Graph[1][4] = 7;  
Graph[1][5] = 3;  
Graph[1][6] = 7;

Graph[2][0] = 1;  
Graph[2][1] = 2;  
Graph[2][2] = 0;  
Graph[2][3] = 1;  
Graph[2][4] = 3;  
Graph[2][5] = 0;  
Graph[2][6] = 0;

Graph[3][0] = 2;  
Graph[3][1] = 0;  
Graph[3][2] = 1;  
Graph[3][3] = 0;  
Graph[3][4] = 0;  
Graph[3][5] = 0;  
Graph[3][6] = 1;

Graph[4][0] = 0;  
Graph[4][1] = 0;  
Graph[4][2] = 3;  
Graph[4][3] = 0;  
Graph[4][4] = 8;  
Graph[4][5] = 1;  
Graph[4][6] = 0;

```
Graph[5][0] = 2;
```

```
Graph[5][1] = 3;
```

```
Graph[5][2] = 8;
```

```
Graph[5][3] = 6;
```

```
Graph[5][4] = 7;
```

```
Graph[5][5] = 2;
```

```
Graph[5][6] = 1;
```

```
Graph[6][0] = 8;
```

```
Graph[6][1] = 4;
```

```
Graph[6][2] = 0;
```

```
Graph[6][3] = 9;
```

```
Graph[6][4] = 0;
```

```
Graph[6][5] = 8;
```

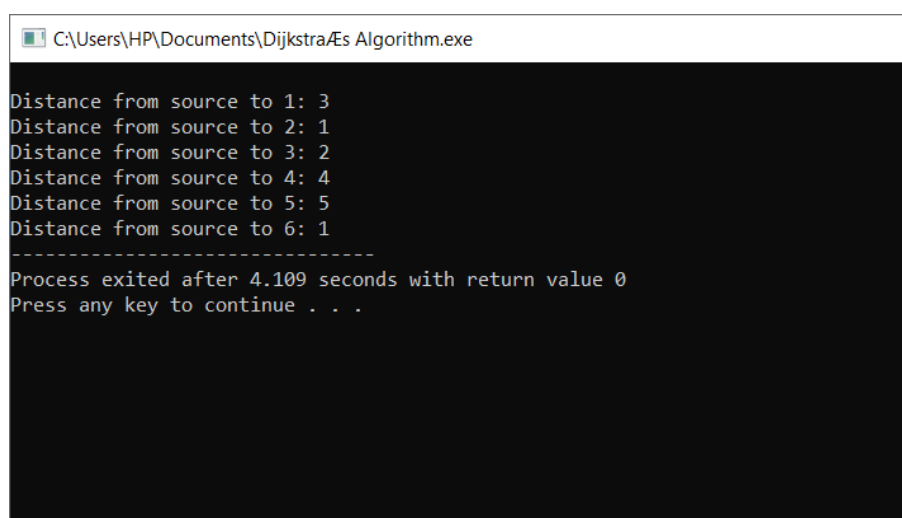
```
Graph[6][6] = 0;
```

```
u = 0;
```

```
Dijkstra(Graph, n, u);
```

```
return 0;
```

```
}
```



```
C:\Users\HP\Documents\Dijkstra\Es Algorithm.exe
Distance from source to 1: 3
Distance from source to 2: 1
Distance from source to 3: 2
Distance from source to 4: 4
Distance from source to 5: 5
Distance from source to 6: 1
-----
Process exited after 4.109 seconds with return value 0
Press any key to continue . . .
```