NAME:	Vedanti Anil Wadatkar
UID:	2021700071
SUBJECT	Design and Analysis of Algorithm
EXPERIMENT NO :	06
AIM:	To find shortest path using Dijkstra's Algorithm.
PROBLEM STATEMENT 1:	**shortest path using Dijkstra's Algorithm and prim's algorithm.**
ALGORITHM and THEORY:	**function** Dijkstra(*Graph*, *source*): 2 3 **for each** vertex *v* in *Graph.Vertices*: 4 dist[*v*] ← INFINITY 5 prev[*v*] ← UNDEFINED *6* add *v* to *Q* 7 dist[*source*] ← 0 8 9 **while** *Q* is not empty: 10 *u* ← vertex in *Q* with min dist[u] *11* remove u from *Q* 12 13 **for each** neighbor *v* of *u* still in *Q*: 14 *alt* ← dist[*u*] + Graph.Edges(*u*, *v*) 15 **if** *alt* < dist[*v*]: *16* dist[*v*] ← *alt*
	17 prev[*v*] ← *u* 18 19 **return** dist[], prev[]
PROGRAM:	#include <limits.h> #include <stdbool.h>

```c
#include <stdio.h>
int V;
int minDistance(int dist[], bool sptSet[])
{
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++)
        if (sptSet[v] == false && dist[v] <= min)
            min = dist[v], min_index = v;
    return min_index;
}

void printSolution(int dist[])
{
    printf("Vertex \t\t Distance from Source\n");
    for (int i = 0; i < V; i++)
        printf("%d \t\t\t %d\n", i, dist[i]);
}

void dijkstra(int graph[V][V], int src)
{
    int dist[V];
    bool sptSet[V];

    for (int i = 0; i < V; i++)
        dist[i] = INT_MAX, sptSet[i] = false;

    dist[src] = 0;
```

```c
    for (int count = 0; count < V - 1; count++)
      { int u = minDistance(dist, sptSet);

      sptSet[u] = true;

      for (int v = 0; v < V; v++)

          if (!sptSet[v] && graph[u]
             [v] && dist[u] !=
             INT_MAX
             && dist[u] + graph[u][v] <
             dist[v]) dist[v] = dist[u] + graph[u]
             [v];
    }
    printSolution(dist);
}

int main()
{
    printf("Enter the
    order:");
    scanf("%d",&V);
    int graph[V][V];
     for(int i=0;i<V;i+
     +)
     {
        printf("Elememts of row number %d:",
        (i+1)); for(int j=0;j<V;j++)
        {
           scanf("%d",&graph[i][j]);
        }
     }
     dijkstra(graph, 0);
     return 0;
}
```

OUTPUT:	
	``
students@students-HP-280-G3-MT:~$ gcc Dijsktra.c
students@students-HP-280-G3-MT:~$ ./a.out
Enter the order:9
Elememts of row number 1:0 4 0 0 0 0 0 8 0
Elememts of row number 2:4 0 8 0 0 0 0 11 0
Elememts of row number 3:0 8 0 7 0 4 0 0 2
Elememts of row number 4:0 0 7 0 9 14 0 0 0
Elememts of row number 5:0 0 0 9 0 10 0 0 0
Elememts of row number 6:0 0 4 14 10 0 2 0 0
Elememts of row number 7:0 0 0 0 0 2 0 1 6
Elememts of row number 8:8 11 0 0 0 0 1 0 7
Elememts of row number 9:0 0 2 0 0 0 6 7 0
Vertex   Distance from Source
0   0
1   4
2   12
3   19
4   21
5   11
6   9
7   8
8   14
students@students-HP-280-G3-MT:~$
``` 

```
students@students-HP-280-G3-MT:~$ gcc Dijsktra.c
students@students-HP-280-G3-MT:~$ ./a.out
Enter the order:4
Elememts of row number 1:2 3 5 1
Elememts of row number 2:5 2 3 7
Elememts of row number 3:2 9 3 5
Elememts of row number 4:1 4 6 9
Vertex   Distance from Source
0   0
1   3
2   5
3   1
students@students-HP-280-G3-MT:~$
``` |
| | |

CONCLUSION:	By performing the above experiment i have successfully found the shortest part of different vertices from a single source using Dijkstra's algorithm.