Revanth. NM
1BM18CS081

Shortest Path using Dijkstra's Algorithm.

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <math.h>
#define IN 99
#define N 6

int dijsktra (int cost[][N], int source, int target)
{
    int dist[N], prev[N], selected[N] = {0}, i, m, min, start, d, j;
    char path[N];
    for (i=1; i<N; i++){
        dist[i] = IN;
        prev[i] = -1;
    }
    start= source;
    selected[start] = 1;
    dist[start]=0;
    while (selected[target]==0){
        min = IN;
        m=0;
        for(i=1; i<N; i++){
            d = dist[start] + cost[start][i];
            if (d< dist[i] && selected[i]==0){
                dist[i] = d;
                prev[i] = start;
            }
            if (min>dist[i] && selected[i]==0){
                min = dist[i];
                m= i;
            }
        }
```

Revanth .NM
1BM18CS081

```
    start = m;
    selected [start] = 1;
  }
  start = target;
  j=0;
  while (start != -1){
     path [j++] = start +65;
     start = prev [start];
  }
  Path [j] = '\0';
  strrev (path);
  printf ("%s", path);
  return dist[target];
}

int main(){
  int cost [N][N], i, j, w, ch, co, source, target, x, y;
  Printf ("\t Shortest path ");
  for (i=1; i<N ; i++)
     for (j=1 ; j<N; j++)
          cost[i][j]= IN;
  for( x=1; x<N ; x++){
     for(y=x+1; y<N ; y++){
        Printf (" Enter the weight of the path b/w %d and %d", x, y);
        scanf ("%d", &w);
        cost [x][y]= cost[y][x] = w;
     }
     printf ("\n");
  }
  Printf ("\n Enter The source ");
  scanf ("%d", & source);
  printf ("\n Enter the target ");
  scanf ("%d", & target);
  printf ("\n Shortest path: %d", dijsktra(cost, source, target));
}
```

Revanth .NM