Implement Dijkstra's algorithm

Shalini B
1BM19CS406

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<math.h>
#define IN 99
#define N 6
int dijkshistra(int cost[][N], int source, int target);
int dijkshtra(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;
            }
        }
```

```
        start = m;
        selected[start] = 1;
    }
    start = target;
    j = 0;
    while(start != -1)
    {
        path[j++] = start + 65;
        start = prev[start];
    }
    path[j] = '\0';
    strrev(path);
    cout << path << endl;
    return dist[target];
}
int main()
{
    int cost[N][N], i, j, w, ch, co;
    int source, target, x, y;
    cout << "Shortest Path Algorithm" << endl;
    for(i = 1; i < N; i++)
        for(j = 1; j < N; j++)
            cost[i][j] = INF;
            for(x = 1; x < N; x++)
            {
                for(y = x + 1; y < N; y++)
                {
                    cout << "Enter weight of path b/w node " << x << y;
                    cin >> w;
                    cost[x][y] = cost[y][x] = w;
                }
                cout << endl;
            }
            cout << "Enter the source";
            cin >> source;
            cout << "Enter the target"; cin >> target;
            co = dijkstra(cost, source, target); cout << "Shortest path" << co;
}
```