Exp (25): implementation of shortest path Algorithms
using Dijkstra's Algorithm

Aim: To implementation of shortest path Algorithms
using Dijkstra's Algorithm

Algorithm:

* stack.
* input no of vertices and adjacency matrix.
* set distance of source to 0, others to infinity.
* find unvisited node with smallest distance.
* update distances of adjacent nodes.
* Repeat until all nodes are visited.
* print shortest distances.
* stop

program:

```c
#include <stdio.h>
#define V 5
#define INF 999
int main() {
    int g[V][V] = {
        { 0, 10, 0, 30, 100},
        { 10, 0, 50, 0, 0},
        { 0, 50, 0, 20, 10},
        { 30, 0, 20, 0, 60},
        { 100, 0, 10, 60, 0}
    };
    int dist[V], vis[V] = {0}, u;
    for(int i=0; i<V; i++) dist[i] = INF;
    dist[0] = 0;
    for(int c=0; c<N-1; c++){
        int min = INF;
        for(int i=0; i<V; i++)
            if(!vis[i] && dist[i]<min){ min=dist[i];
                                        u=i;}
        vis[u] = 1;
        for(int v=0; v<V; v++)
```

```
if ( g[u][v] && dist [v] > dist [u] + g[u][v])
        dist [v] = dist [u] + g[u][v];
}
for (int i = 0; i < v; i++)
        printf(" 0 + %d = %d \n", i, dist [i]);
}
```

output:

```
0 → 0 = 0
0 → 1 = 10
0 → 2 = 50
0 → 3 = 30
0 → 4 = 60
```

RESULT: Thus, the program executed successfully.