

12/01/23

Date _____
Page _____

Q9 implement Dijkstra's Algo to compute the shortest path from a given
Dijkstra's Algorithm

```
#include <bits/stdc++.h>
```

```
#include <limits.h>
```

```
#include <stdio.h>
```

```
using namespace std;
```

```
#define V 9
```

```
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 & Distance from source\n");
```

```
    for (int i = 0; i < V; i++)
```

```
        printf("%d\t\t%d\n", i, dist[i]);
```

```
}
```

for a given
graph.

```
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;
```

```
for (int count=0; count<V; count++)
```

```
{
    int u = minDist (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];
}
```

```
printAns (dist);
```

```
}
```

```
int main()
```

```
{
```

```
int graph [V][V];
```

```
read n // with graph & edges
```

28/01/23

$f_u(\{u, v\} \in U(i, h))$

{

$f_u(\{u, v\} \in U(i, h))$

$\rightarrow \text{graph}(G(i, h))$

}

$\text{dist}(u, v)$

return 0;

}

o/p

Enter graph

0 9 2 5

9 0 6 8

2 6 0 0

5 8 0 0

Value

0

1

2

3

Distance from Sam

0

8

2

5