NAME:	Pratham Jain
UID:	2021300051
SUBJECT	Design and Analysis of Algorithm
EXPERIMENT NO :	6A
DATE OF PERFORMANCE	27/03/2023
DATE OF SUBMISSION	03/04/2023
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:	

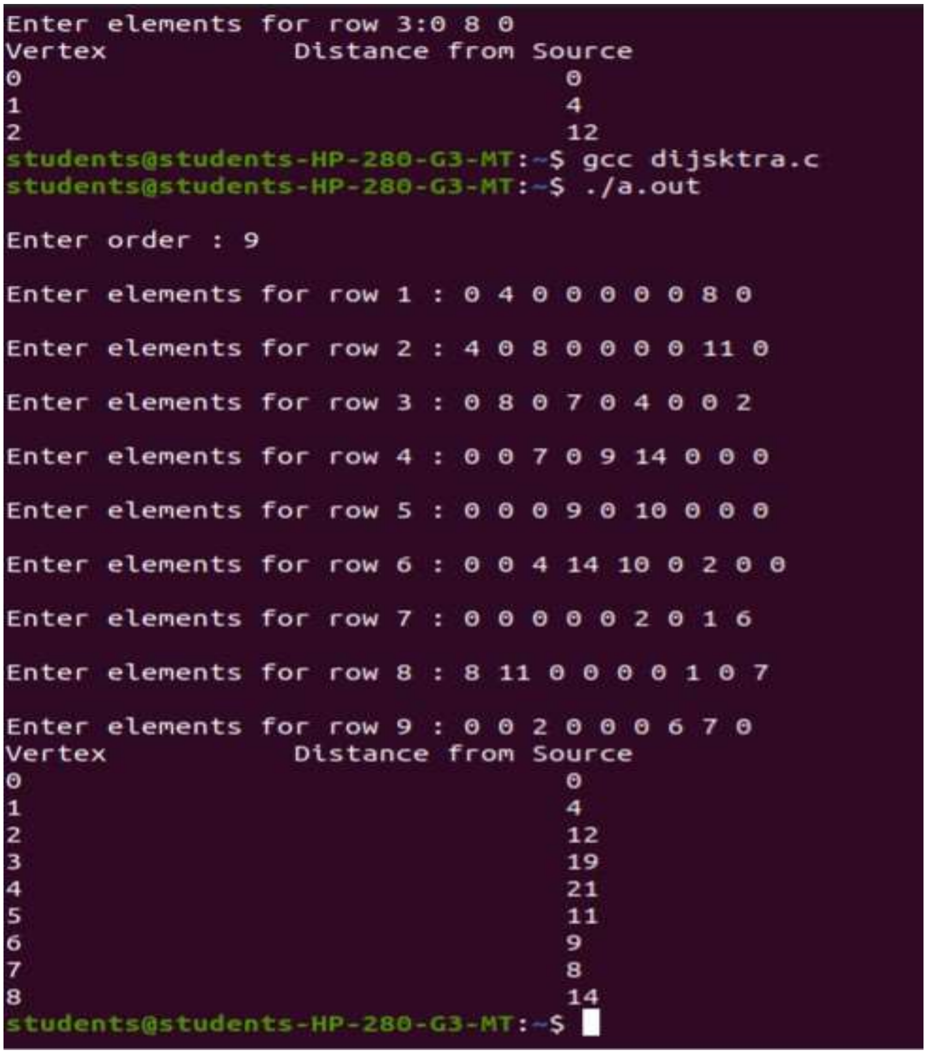
```
function Dijkstra(Graph, source):

    for each vertex v in Graph.Vertices:
        dist[v] ← INFINITY
        prev[v] ← UNDEFINED
        add v to Q
    dist[source] ← 0

    while Q is not empty:
        u ← vertex in Q with min dist[u]
        remove u from Q

        for each neighbor v of u still in Q:
            alt ← dist[u] + Graph.Edges(u, v)
            if alt < dist[v]:
                dist[v] ← alt
            prev[$v$] ← $u$

    **return** dist[], prev[]
```

PROGRAM:

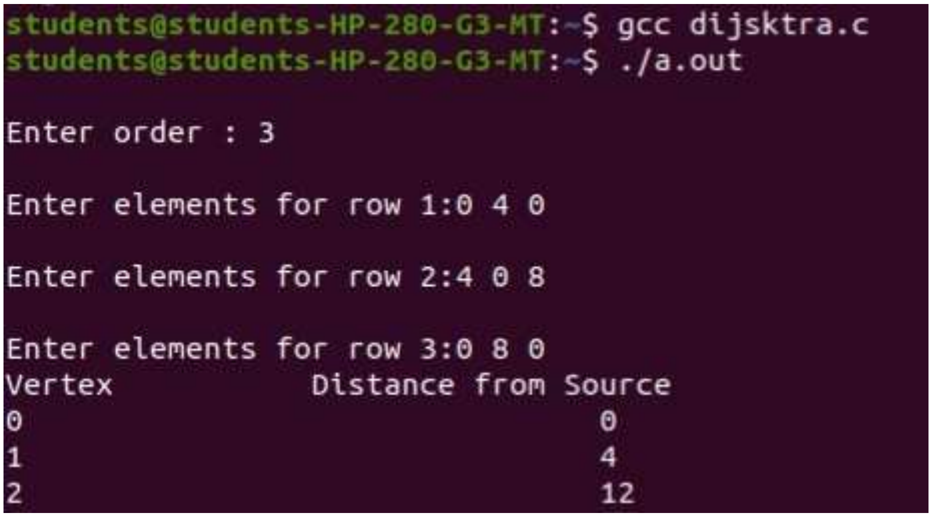
```
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
int V;
int minDistance(int dist[], bool sptSet[])
{
	int min = INT_MAX, min_index;

	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\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;
	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("\nEnter order : ");
    scanf("%d",&V);
    int graph[V][V];
    for(int i=0;i<V;i++)
    {
        printf("\nEnter elements for row %d : ",(i+1));
        for(int j=0;j<V;j++)
        {
            scanf("%d",&graph[i][j]);
        }
    }
    dijkstra(graph, 0);
    return 0;
}
```

OUTPUT:



```
Enter elements for row 3:0 8 0
Vertex           Distance from Source
0                                0
1                                4
2                                12
students@students-HP-280-G3-MT:~$ gcc dijsktra.c
students@students-HP-280-G3-MT:~$ ./a.out

Enter order : 9

Enter elements for row 1 : 0 4 0 0 0 0 0 8 0

Enter elements for row 2 : 4 0 8 0 0 0 0 11 0

Enter elements for row 3 : 0 8 0 7 0 4 0 0 2

Enter elements for row 4 : 0 0 7 0 9 14 0 0 0

Enter elements for row 5 : 0 0 0 9 0 10 0 0 0

Enter elements for row 6 : 0 0 4 14 10 0 2 0 0

Enter elements for row 7 : 0 0 0 0 0 2 0 1 6

Enter elements for row 8 : 8 11 0 0 0 0 1 0 7

Enter elements for row 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 order : 3

Enter elements for row 1:0 4 0

Enter elements for row 2:4 0 8

Enter elements for row 3:0 8 0
Vertex           Distance from Source
0                                0
1                                4
2                                12
```

CONCLUSION:	I have successfully understood and implemented the concept of Dijkstra’s Algorithm through ths experiment. I was also able to understand how it helps us to find the shortest path in a matrix.