Name: Vanessa D'mello
Roll No: 8863
Branch: SE Computers A (Batch A)
Experiment: Dijkstra's SSSP Algorithm

```
#include <stdio.h>
void dijkstra(int v, int a[][v], int s, int dist[], int pred[], int visited[]) {
	for(int i = 0; i < v - 1; i++) {
		int min = 9999;
		int min_vertex;
		for(int j = 0; j < v; j++) {
			if(visited[j] == 0 && dist[j] < min) {
				min = dist[j];
				min_vertex = j;
			}
		}
		visited[min_vertex] = 1;
		for(int j = 0; j < v; j++) {
			if(j == s)
				continue;
			if(a[min_vertex][j] != 0 && dist[j] > dist[min_vertex] + a[min_vertex][j])
{
				dist[j] = dist[min_vertex] + a[min_vertex][j];
				pred[j] = min_vertex + 1;
			}
		}
	}
	shortestPath(dist, pred, v, s);
}
void shortestPath(int dist[], int pred[], int v, int s) {
	printf("\nDistance\tPath");
	for(int i = 0; i < v; i++) {
		if(i == s)
			continue;
		printf("\n %d\t\t", dist[i]);
		printf("->%d", getPath(i + 1, s, pred));
	}
}
int getPath(int n, int s, int pred[]) {
	if(n == s + 1)
		return n;
	else if(pred[n - 1] == s + 1)
		printf("%d", getPath(pred[n - 1], s, pred));
	else
		printf("->%d", getPath(pred[n - 1], s, pred));
	return n;
}
```

```
int main() {
	int v, source, dest, w;
	printf("Enter no. of vertices: ");
	scanf("%d", &v);
	int a[v][v];
	for(int i = 0; i < v; i++) {
		for(int j = 0; j < v; j++)
		a[i][j] = 0;
	}
	int e;
	printf("Enter no. of edges: ");
	scanf("%d", &e);
	for(int i = 0; i < e; i++) {
		printf("Enter source and destination of edge %d: ", i + 1);
		scanf("%d %d", &source, &dest);
		printf("Enter weight of edge %d: ", i + 1);
		scanf("%d", &w);
		a[source - 1][dest - 1] = w;
	}
	int s = 1;
	printf("Enter source vertex: ");
	scanf("%d", &s);
	int visited[v];
	int dist[v];
	int pred[v];
	for(int i = 0; i < v; i++) {
		if(i == s - 1) {
			visited[i] = 1;
			pred[i] = i + 1;
		} else {
			visited[i] = 0;
			pred[i] = s;
		}
		if(a[s - 1][i] || i == s - 1)
			dist[i] = a[s - 1][i];
		else
			dist[i] = 9999;
	}
	dijkstra(v, a, s - 1, dist, pred, visited);
	printf("\nDist: ");
	for(int i = 0; i < v; i++)
		printf("%d ", dist[i]);
	printf("\nPred: ");
	for(int i = 0; i < v; i++)
		printf("%d ", pred[i]);
	printf("\nVisited: ");
	for(int i = 0; i < v; i++)
```

```
            printf("%d ", visited[i]);
    return 0;
}
```

Postlab Output: