```
#include <stdio.h>

#define infinity 999

void dijkstra(int n, int v, int cost[20][20], int dist[]) {
   int i, u, count, w, flag[20], min;

   for (i = 1; i <= n; i++)
      flag[i] = 0, dist[i] = cost[v][i];

   count = 2;

   while (count <= n) {
      min = infinity;

      for (w = 1; w <= n; w++)
         if (dist[w] < min && !flag[w]) {
            min = dist[w];
            u = w;
         }

      flag[u] = 1;
      count++;

      for (w = 1; w <= n; w++)
         if ((dist[u] + cost[u][w] < dist[w]) && !flag[w])
            dist[w] = dist[u] + cost[u][w];
   }
}
```

```
int main() {
    int n, v, i, j, cost[20][20], dist[20];

    printf("Enter the number of nodes: ");
    scanf("%d", &n);

    printf("\nEnter the cost matrix:\n");
    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++) {
            scanf("%d", &cost[i][j]);
            if (cost[i][j] == 0)
                cost[i][j] = infinity;
        }

    printf("\nEnter the source matrix: ");
    scanf("%d", &v);

    dijkstra(n, v, cost, dist);

    printf("\nShortest path:\n");
    for (i = 1; i <= n; i++)
        if (i != v)
            printf("%d->%d, cost=%d\n", v, i, dist[i]);

    return 0;
}
```