From a given vertex in a weighted connected graph, find shortest paths to other vertices

using Dijkstra's algorithm.

```
#include <stdio.h>

int main() {
   int i, j, n, v, k, min, u, c[20][20], s[20], d[20];

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

   printf("Enter the cost adjacency matrix (999 for no edge):\n");
   for (i = 1; i <= n; i++) {
      for (j = 1; j <= n; j++) {
         scanf("%d", &c[i][j]);
      }
   }

   printf("Enter the source vertex: ");
   scanf("%d", &v);

   for (i = 1; i <= n; i++) {
      s[i] = 0;
      d[i] = c[v][i];
   }

   d[v] = 0;
   s[v] = 1;

   for (k = 2; k <= n; k++) {
```

```
        min = 999;
        for (i = 1; i <= n; i++) {
            if (!s[i] && d[i] < min) {
                min = d[i];
                u = i;
            }
        }
        s[u] = 1;
        for (i = 1; i <= n; i++) {
            if (!s[i] && d[i] > d[u] + c[u][i]) {
                d[i] = d[u] + c[u][i];
            }
        }
    }

    printf("The shortest distances from vertex %d are:\n", v);
    for (i = 1; i <= n; i++) {
        printf("%d --> %d = %d\n", v, i, d[i]);
    }

    return 0;
}
```

Output

```
Enter the number of vertices: 2
Enter the cost adjacency matrix (999 for no edge):
2
999
666
333
Enter the source vertex: 4
The shortest distances from vertex 4 are:
4 --> 1 = -1347155110
4 --> 2 = -1347155776


=== Code Execution Successful ===
```