Exp 4: Implement Dijikstra's algorithm to compute the shortest path for a given topology.

Program:

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

int c[10][10], n, src;
void dijikstra();
int main ()
 {
      printf("\n enter the number of vertices \n");
      scanf("%d", &n);
      printf("\n enter the cost matrix \n");
      for (int i=1; i<=n; i++)
       {
          for (int j=1; j<=n; j++)
           {
             scanf ("%d", &c[i][j]);
           }
       }
       printf ("\n enter the source veton \n");
       scanf ("%d", &src);
       dijikstra ();
       return 1;
 }
void dijikstra ()
{
int dist [10], vis [10], j, count, min, u;
for (j=1; j<=n; j++)
 {
      dist [j] = c[sr] [j];
 }
for (j=1;j<=n;j++)
{
  vis [j] = 0;
}
  dis [src] = 0;
  vis [src]=1;
  count =1;
  while (count ! =n)
   {
      min =9999;
         for (j=1; j<=n;j++)
         {
            if (dist [j]<min && vis [j]! =1)
            {
               min = dist [j];
                  u=j ;
            }
         }
  vis [u]=1;
  count ++;
  for (j=1;j<=n;j++)
   {
      if (min + c[u] [j] < dist [j] && vis [j] ! =1)
        {
          dist [j] = min + c [u] [j];
        }
      }
   }
  printf ("\n shortest distance is \n");
   for (j=1;j<=n;j++)
   {
     printf ("\n %d -----> %d =%d\n",src,j,dist[j]);
   }
}
```

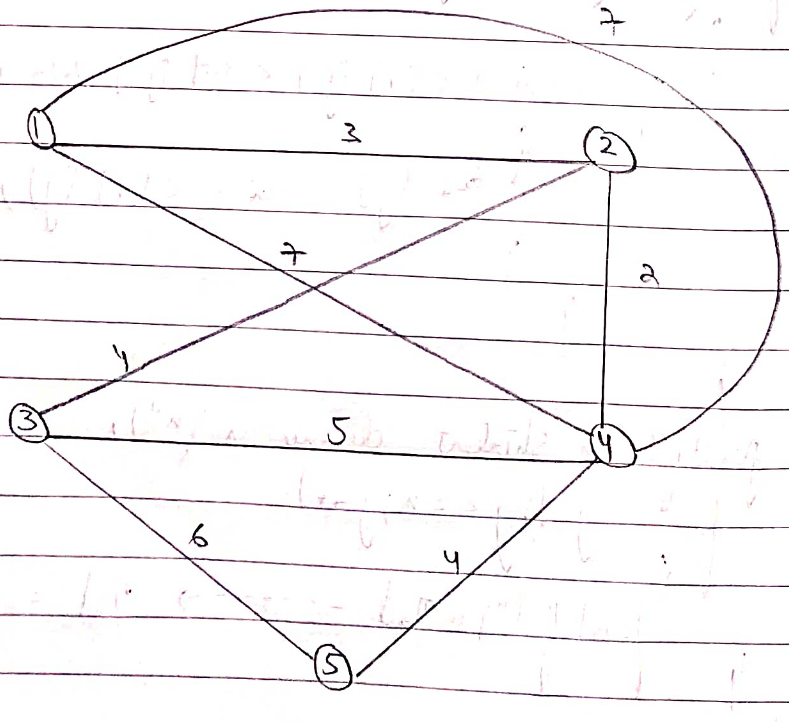
OUTPUT:

```
Enter the number of vertices
5

Enter the cost matrix
9999   3   9999  7   9999
3   9999   4   2   9999
9999   4   9999  5   6
7   2   5   9999   4
9999   9999   6  4   9999

Enter the source vertex
1

Shortest distance is
1 ------> 1 = 0
1 ------> 2 = 3
1 -------> 3 = 7
1 -------> 4 = 5
1 --------> 5 = 9.
```


7
1
3
2
7
2
4
3
5
4
6
4
5