Program 9

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

void dijs(int n,int sv, int cost[10][10],int dist[],int pred[])
{
      int i,v,count,w,j,visited[23],min;
      for (i=0;i<n;i++)
      {
            visited[i]=0;
            dist[i]=cost[sv][i];
            if( dist[i] !=999)
               pred[i]=sv;
      }

      visited[sv]=1;
      dist[sv]=0;
      pred[sv]=-1;
      count=1;
      while (count< n)
      {
            min=999;
            for (w=0;w<n;w++)
            {
                  if (!visited[w] && dist[w]<min  )
                  {
                        min =dist[w];
                        v=w;
                  }
            }

            visited[v]=1;
            count++;

            for (w=0;w<n;w++)
            {
                  if (!visited[w] &&  dist[v]+cost[v][w]<dist[w])
                      {
                              dist[w]=dist[v]+cost[v][w];
                              pred[w]=v;
```

```
                        }
                }
        }

}

main()
{

// weight of an edge if 999 means infinity : vertices are not connected
        int n,sv,i,j,dist[10],cost[10][10],pred[10];
        printf("\nDijkstra\n");
        printf("Enter the number of vertices:");
        scanf("%d",&n);
        printf("Enter the cost of matrix:\n");
        for (i=0;i<n;i++)
        {
                for (j=0;j<n;j++)
                {
                        scanf("%d",&cost[i][j]);
                        if (cost[i][j]==0)
                                cost[i][j]=999;
                }
        }
        printf("Enter the source:");
        scanf("%d",&sv);
        dijs(n,sv,cost,dist,pred);
        printf("\nShortest path:\n");

        for(i=0;i<n;i++)
          if(i!=sv)
          {
            printf("\nShotest Distance of vertex %d=%d",i,dist[i]);
            printf("\nPath=%d",i);
                j=i;
            do
            {
               j=pred[j];
               printf("<-%d",j);
             }while(j!=sv);
          }
}
```