1)

```
#include<stdio.h>
#define MAX 10
#define INFINITY 9999
#define CNT count++
int GRAPH[MAX][MAX];
int V,E;
enum STATUS{TRUE,FALSE};
int count =0;


void G_create()
{
  int i,j;
  printf("ENTER NUMBER OF VERTICES:\n");
  scanf("%d",&V);

  printf("ENTER ADJACENCY MATRIX:\n");

  for(i=0;i<V;i++)
  for(j=0;j<V;j++)
  {
      scanf("%d",&GRAPH[i][j]);
      if(i!=j&&GRAPH[i][j]==0)
      GRAPH[i][j]=INFINITY;
       if(i==j)
      GRAPH[i][j]=0;
  }

}

void dijkstras()
{
    enum STATUS S[MAX];
    int i,j,src,index,w,dist[MAX];

    printf("ENTER SOURCE VERTEX: ");
    scanf("%d",&src);
     printf("\n\nSINGLE SOURCE SHORTEST PATH (VERTEX:
DISTANCE)\n");
    for(i=0;i<V;i++)
    {
        CNT; dist[i]=GRAPH[src][i];CNT;
        S[i]=FALSE;CNT;
    }
    dist[src]=0;CNT;
    S[src]=TRUE;CNT;

    for(i=0;i<V-1;i++)
    {
        CNT;int min=INFINITY;CNT;
        for(j=0;j<V;j++)
        {
           CNT;
            if(dist[j]<=min && S[j]==FALSE)
            {
```

```
                CNT; min=dist[j];CNT;
                 index=j;CNT;
             }
        }

        S[index]=TRUE;CNT;

        for(w=0;w<V;w++)
        {
           CNT;
            if(S[w]==FALSE && dist[w]>(dist[index]+GRAPH[index][w])
&& dist[index]!=INFINITY && GRAPH[index][j]!=INFINITY)
            {CNT;dist[w]=dist[index]+GRAPH[index][w];CNT;}
        }

    }
    for(i=0;i<V;i++)
    {
        if(dist[i]==INFINITY)
        printf("%d: NO PATH!\n",i);
        else
        printf("%d: %d\n",i,dist[i]);
    }

}

int main()
{
    G_create();
    dijkstras();
    printf("STEP COUNT: %d\n",count);

}
```

OUTPUT

```
"C:\Users\Lloyd\Documents\C-Free\Temp\Untitled19.exe"
ENTER NUMBER OF VERTICES:
8
ENTER ADJACENCY MATRIX:
0 0 0 0 0 0 0 0
300 0 0 0 0 0 0 0
1000 800 0 0 0 0 0 0
0 0 1200 0 0 0 0 0
0 0 0 1500 0 250 0 0
0 0 0 1000 0 0 900 1400
0 0 0 0 0 0 1000 0
1700 0 0 0 0 0 0 0
ENTER SOURCE VERTEX: 3


SINGLE SOURCE SHORTEST PATH (VERTEX: DISTANCE)
0: 2200
1: 2000
2: 1200
3: 0
4: NO PATH!
5: NO PATH!
6: NO PATH!
7: NO PATH!
STEP COUNT: 211
Press any key to continue . . .
```

1)

```
#include<stdio.h>
#define MAX 10
#define INFINITY 999
#define NEXTLINE printf("\n")
#define CNT count++
#define Print_Infinity     printf("oo   ")
int GRAPH[MAX][MAX],A[MAX][MAX];
int V,E;
int count=0;

void G_create()
{
  int i,j;
  printf("ENTER NUMBER OF VERTICES:\n");
  scanf("%d",&V);

  printf("ENTER ADJACENCY MATRIX:\n");

  for(i=0;i<V;i++)
  for(j=0;j<V;j++)
  {
      scanf("%d",&GRAPH[i][j]);
      if(i!=j&&GRAPH[i][j]==0)
      GRAPH[i][j]=INFINITY;
       if(i==j)
      GRAPH[i][j]=0;
  }

}

void Reset()
{
      int i,j;
      for(i=0;i<V;i++)
      {
            for(j=0;j<V;j++)
            {
                  if(GRAPH[i][j]==0&&i!=j)
                  GRAPH[i][j]=INFINITY;
                  if(i==j)
                  GRAPH[i][j]=0;
            }
      }
}

int min(int a, int b)
{
      if(a<b)
      return a;
      else
      return b;

}

void Line_Generator(int n)
```

```
{
      int i;
      for(i=0;i<n;i++)
      printf("-");
}

void print_G(int x)
{
  int i,j;
      NEXTLINE;NEXTLINE;
  printf(" A%d |   ",x);
   for(i=0;i<V;i++)
        printf("%-3d  ",i);
        NEXTLINE;Line_Generator((V*5)+5);NEXTLINE;  /*NO. OF -
BETWEEN 2 ROWS IS 5 DASHES AND INTIALLY 5 PREDEFINED DASHES*/
  for(i=0;i<V;i++)
  {
       printf("%-2d  |   ",i);

       for(j=0;j<V;j++)
       {
              if(A[i][j]==INFINITY)
              Print_Infinity;
              else
                  printf("%-3d  ",A[i][j]);
       }

       printf("\n");
  }
}

void All_Pair_Shortest()
{
      int i,j,k;
      for(i=0;i<V;i++)
      {
            CNT;
            for(j=0;j<V;j++)
            {CNT;A[i][j]=GRAPH[i][j];CNT;}
      }
      print_G(0);

      for(k=0;k<V;k++)
      {
            CNT;
            for(i=0;i<V;i++)
            {
                  CNT;
                  for(j=0;j<V;j++)
                  {CNT;A[i][j]=min(A[i][j],A[i][k]+A[k][j]);CNT;}
            }
            print_G(k+1);
      }

 }
```

```
int main()
{
      G_create();
      Reset();
      All_Pair_Shortest();
      printf("\nSTEP COUNT: %d\n",count);
}
```

OUTPUT

```
"C:\Users\Lloyd\Documents\C-Free\Temp\Untitled20.exe"
ENTER NUMBER OF VERTICES:
3
ENTER ADJACENCY MATRIX:
0 4 11
6 0 2
3 0 0


 A0 |   0    1    2
--------------------
0   |   0    4    11
1   |   6    0    2
2   |   3    oo   0


 A1 |   0    1    2
--------------------
0   |   0    4    11
1   |   6    0    2
2   |   3    7    0


 A2 |   0    1    2
--------------------
0   |   0    4    6
1   |   6    0    2
2   |   3    7    0


 A3 |   0    1    2
--------------------
0   |   0    4    6
1   |   5    0    2
2   |   3    7    0

STEP COUNT: 87
Press any key to continue . . .
```