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()

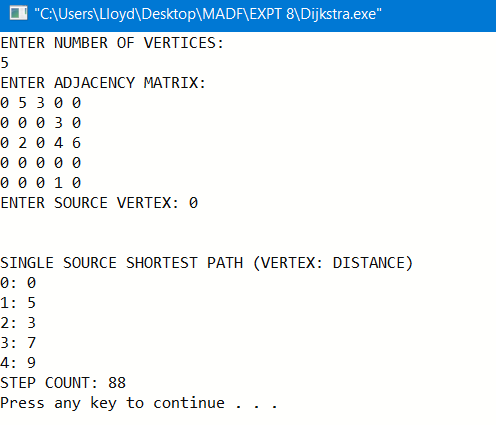
{

G\_create();

dijkstras();

printf("STEP COUNT: %d\n",count);

}



OUTPUT

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

