**Warshalls Algorithm**

#include<stdio.h>

#include<conio.h>

#include<math.h>

int max(int,int);

void warshal(int p[10][10],int n)

{

 int i,j,k;

 for(k=1;k<=n;k++)

  for(i=1;i<=n;i++)

   for(j=1;j<=n;j++)

    p[i][j]=max(p[i][j],p[i][k]&&p[k][j]);

}

int max(int a,int b)

{                                                       ;

if(a>b)

 return(a);

else

 return(b);

}

void main()

{

 int p[10][10]={0},n,e,u,v,i,j;

 clrscr();

 printf("\n Enter the number of vertices:");

 scanf("%d",&n);

 printf("\n Enter the number of edges:");

 scanf("%d",&e);

 for(i=1;i<=e;i++)

 {

  printf("\n Enter the end vertices of edge %d:",i);

  scanf("%d%d",&u,&v);

  p[u][v]=1;

 }

 printf("\n Matrix of input data: \n");

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

   printf("%d\t",p[i][j]);

  printf("\n");

 }

 warshal(p,n);

 printf("\n Transitive closure: \n");

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

   printf("%d\t",p[i][j]);

  printf("\n");

 }

 getch();}

**Floyds Algorithm**

#include<stdio.h>

#include<conio.h>

int min(int,int);

void floyds(int p[10][10],int n)

{

 int i,j,k;

 for(k=1;k<=n;k++)

  for(i=1;i<=n;i++)

   for(j=1;j<=n;j++)

    if(i==j)

     p[i][j]=0;

    else

     p[i][j]=min(p[i][j],p[i][k]+p[k][j]);

}

int min(int a,int b)

{

 if(a<b)

  return(a);

 else

  return(b);

}

void main()

{

 int p[10][10],w,n,e,u,v,i,j;;

 clrscr();

 printf("\n Enter the number of vertices:");

 scanf("%d",&n);

 printf("\n Enter the number of edges:\n");

 scanf("%d",&e);

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

   p[i][j]=999;

 }

 for(i=1;i<=e;i++)

 {

  printf("\n Enter the end vertices of edge%d with its weight \n",i);

  scanf("%d%d%d",&u,&v,&w);

  p[u][v]=w;

 }

 printf("\n Matrix of input data:\n");

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

   printf("%d \t",p[i][j]);

  printf("\n");

 }

 floyds(p,n);

 printf("\n Transitive closure:\n");

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

   printf("%d \t",p[i][j]);

  printf("\n");

 }

 printf("\n The shortest paths are:\n");

 for(i=1;i<=n;i++)

  for(j=1;j<=n;j++)

  {

   if(i!=j)

    printf("\n <%d,%d>=%d",i,j,p[i][j]);

  }

 getch();

}

**Prims Algorithm**

#include<stdio.h>

#include<conio.h>

int a,b,u,v,n,i,j,ne=1;

int visited[10]={0},min,mincost=0,cost[10][10];

void main()

{

 clrscr();

 printf("\n Enter the number of nodes:");

 scanf("%d",&n);

 printf("\n Enter the adjacency matrix:\n");

 for(i=1;i<=n;i++)

  for(j=1;j<=n;j++)

  {

   scanf("%d",&cost[i][j]);

   if(cost[i][j]==0)

    cost[i][j]=999;

  }

 visited[1]=1;

 printf("\n");

 while(ne<n)

 {

  for(i=1,min=999;i<=n;i++)

   for(j=1;j<=n;j++)

    if(cost[i][j]<min)

     if(visited[i]!=0)

     {

      min=cost[i][j];

      a=u=i;

      b=v=j;

     }

  if(visited[u]==0 || visited[v]==0)

  {

   printf("\n Edge %d:(%d %d) cost:%d",ne++,a,b,min);

   mincost+=min;

   visited[b]=1;

  }

  cost[a][b]=cost[b][a]=999;

 }

 printf("\n Minimun cost=%d",mincost);

 getch();

}