

/*C Program to implement Warshall's Algorithm

Input : 1. No. of vertices in the graph

2. No. of edges in the graph

3. End vertices of each edge

Output : Transitive closure matrix

***/**

```
#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);
```

```
}
```

```
int main() {
```

```
    int p[10][10]= {0},n,e,u,v,i,j;
```

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

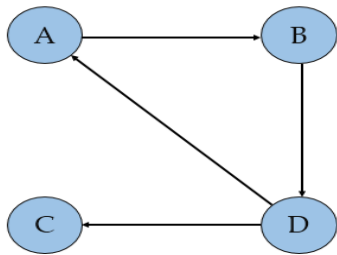
```
    scanf("%d",&n);
```

```
    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();
}
```

Sample Input and Output:



```
Enter the number of vertices:4
Enter the number of edges:4
Enter the end vertices of edge 1:1 2
Enter the end vertices of edge 2:2 4
Enter the end vertices of edge 3:4 3
Enter the end vertices of edge 4:4 1

Matrix of input data:
0 1 0 0
0 0 0 1
0 0 0 0
1 0 1 0

Transitive closure:
1 1 1 1
1 1 1 1
0 0 0 0
1 1 1 1
Press any key to continue..._
```