

```
File Edit Search Run Compile Debug Project Options Window Help  
[ ] TEST.CPP ? [ ]  
include<stdio.h>  
#include<conio.h>  
void main()  
{  
int array1[50],array2[50],array3[100],m,n,i,j,k=0;  
clrscr();  
printf("\nEnter size of array1:");  
scanf("%d",&m);  
printf("Enter sorted elements of array1:\n");  
for(i=0;i<m;i++)  
{  
scanf("%d",&array1[i]);  
}  
printf("\nEnter size of array2:");  
scanf("%d",&n);  
printf("Enter sorted element of array2:\n");  
for(i=0;i<n;i++)  
{  
scanf("%d",&array2[i]);  
}  
i=0;  
1:1 F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

File Edit Search Run Compile Debug Project Options Window Help

TEST.CPP

```
i=0;
j=0;
while(i<m&&j<n)
{
if(array1[i]<array2[j])
{
array3[k]=array1[i];
i++;
}
else
{
array3[k]=array2[j];
j++;
}
k++;
}
if(i>=m)
{
while(j<n)
{
array3[k]=array2[j];
}
```

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

File Edit Search Run Compile Debug Project Options Window Help

```
[ ]= TEST.CPP ?=[ ↑ ]=||

while(i<m)
{
array3[k]=array1[i];
i++;
j++;
}
printf("\n After merging:\n");
for(i=0; i<m+n; i++)
{
printf("\n%d",array3[i]);
}
getch();
}
```

**Enter size of array1:3**

**Enter sorted elements of array1:**

**1 2 3**

**Enter size of array2:3**

**Enter sorted element of array2:**

**5 6 7**

**After merging:**

**1  
2  
3  
5  
6  
7**

```
File Edit Search Run Compile Debug Project Options Window Help  
TEST.CPP 1=[]  
#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("\nEnter the no of nodes:");  
scanf("%d",&n);  
printf("\nEnter 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)  
{  
    1:1  
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

```
File Edit Search Run Compile Debug Project Options Window Help  
TEST.CPP 1=[+] [A]  
{  
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 Minimum cost:%d",mincost);  
    getch();  
}
```

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

Enter the number of nodes:6

Enter the adjacency matrix:

0	4	5	7	8	9
1	0	7	4	3	8
2	4	0	6	8	2
6	4	8	0	7	4
6	7	4	5	0	6
5	4	3	7	4	0

Edge 1: (1 2) cost:4

Edge 2: (2 5) cost:3

Edge 3: (2 4) cost:4

Edge 4: (4 6) cost:4

Edge 5: (6 3) cost:3

Minimum cost 18\_