

Image

The image shows a screenshot of a C programming environment. At the top is a menu bar with options: File, Edit, Search, Run, Compile, Debug, Project, Options, and Window. Below the menu bar, the title bar displays "SORTING.C". The main area contains the following C code:

```
#include<stdio.h>
#include<conio.h>
// C program to accept N numbers and arrange them in ascending order
void main()
{
    int i,j,a,n,number[30];
    clrscr();
    printf("Enter the value of N :");
    scanf("%d",&n);

    printf("Enter the numbers :");
    for (i=0; i<n; ++i)
        scanf("%d",&number[i]);
}
```

The code is intended to sort N numbers in ascending order. It includes headers for stdio.h and conio.h, defines a main function, initializes variables, and uses a for loop to input N numbers into an array.

At the bottom of the screen, there is a terminal window showing the command prompt "1:1 <". Below the terminal window is a keyboard status bar with the following keys: F1 Help, F2 Save, F3 Open, Alt-F9 Compile, F9 Make, F10 Menu.

File Edit Search Run Compile Debug
[] SORTING

```
for (i=0; i<n; ++i)
{
    for (j=i+1; j<n; ++j)
    {
        if (number[i] > number[j])
        {
            a=number[i];
            number[i]=number[j];
            number[j]=a;
        }
    }
}
```

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9

Esc F1 F2 F3 F4 F5 F6 F7 F8
/ @ # \$ % ^ & *

File Edit Search Run Compile Debug Project Options Window

[*]

SORTING.C

```
printf("The numbers arranged in ascending order are given below\n");
for (i=0; i<n; ++i)
    printf("%d\n", number[i]);
getch();
}
```

1:1

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

Enter the value of N :

4

Enter the numbers :

77

11

99

22

The numbers arranged in ascending order are given below

11

22

77

99

-

File Edit Search Run Compile Debug Project

PRIMS1.C

```
#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]);
    }
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile

File Edit Search Run Compile Debug

PRIMS1.C

```
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;

```

28:1

F1 Help Alt-F8 Next Msg Alt-F2 Prev

File Edit Search Run Compile Debug Project

PRIMS1.C

```
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();
}
```

42:1

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile

INSPIRON

Enter the no of nodes:6

Enter the adjacency matrix:

```
0 3 1 6 0 0  
3 0 5 0 3 0  
1 5 0 5 6 4  
6 0 5 0 0 2  
0 3 6 0 0 6  
0 0 4 2 6 0
```

```
edge 1:(13)cost:1  
edge 2:(12)cost:3  
edge 3:(25)cost:3  
edge 4:(36)cost:4  
edge 5:(64)cost:2  
Minimum cost:13
```

DELL



