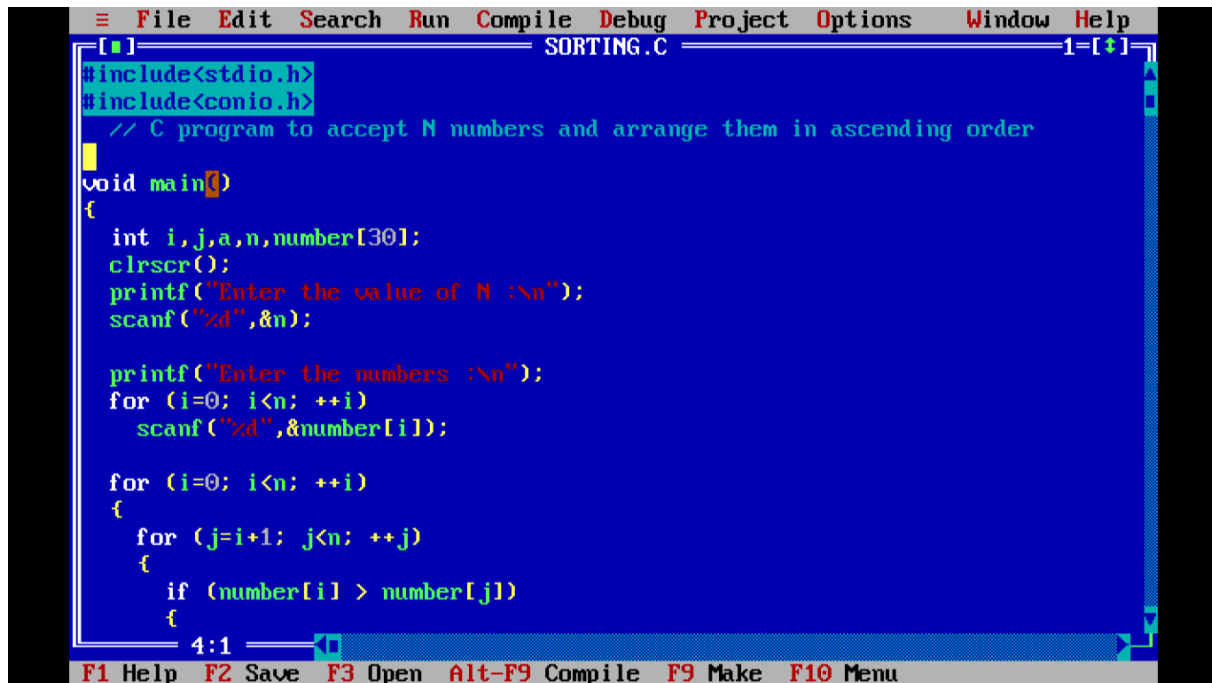


1. Sorting of an integer array

Program



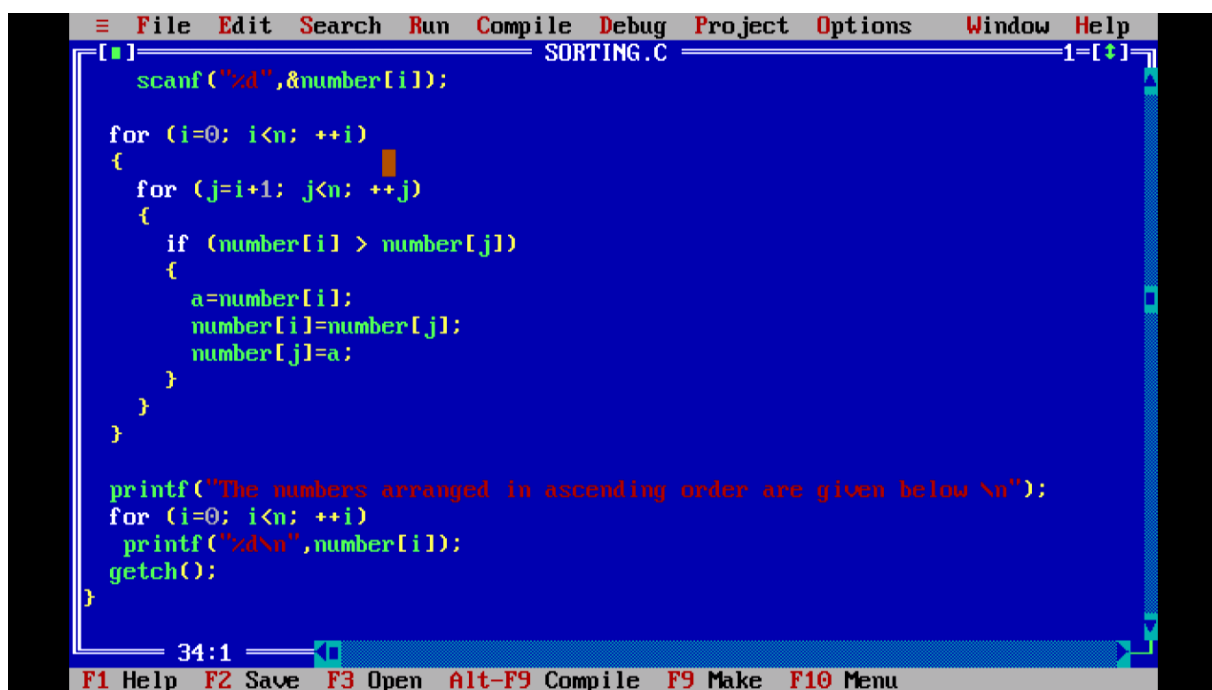
```
File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
#include<stdio.h>
#include<conio.h>
// C program to accept N numbers and arrange them in ascending order

void main()
{
    int i,j,a,n,number[30];
    clrscr();
    printf("Enter the value of N :\n");
    scanf("%d",&n);

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

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

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



```
scanf("%d",&number[i]);

    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;
            }
        }
    }

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

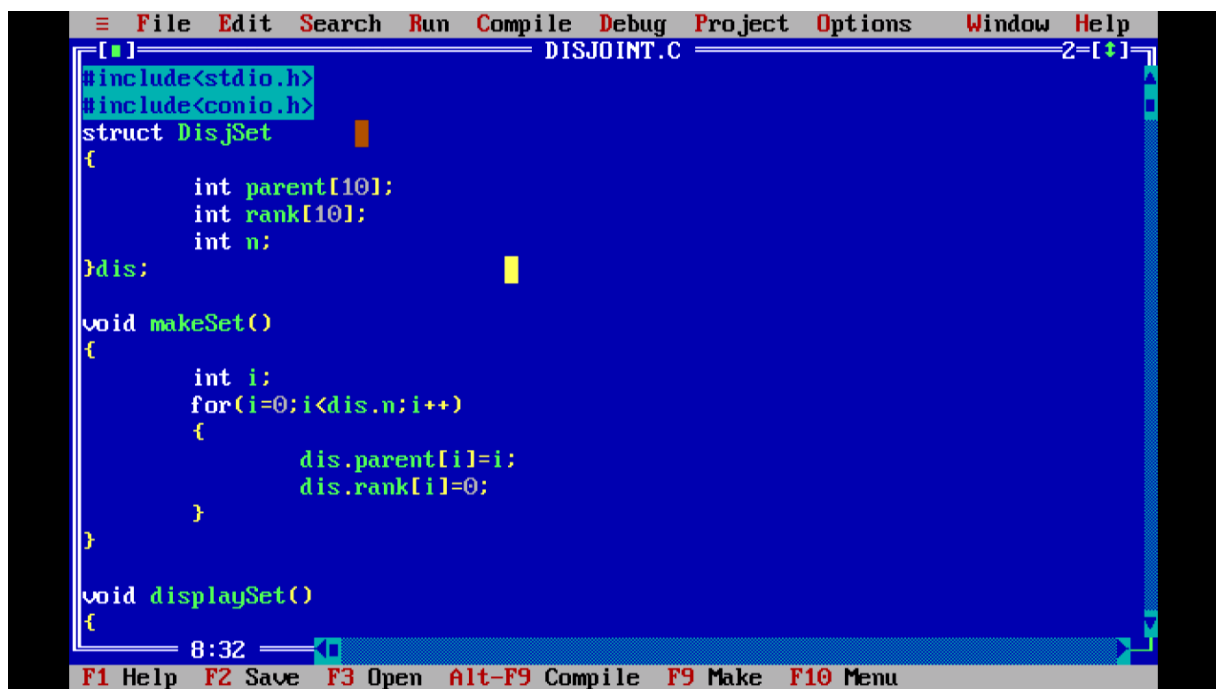
34:1 F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

Output

```
Enter the value of N :
5
Enter the numbers :
6
2
9
12
3
The numbers arranged in ascending order are given below
2
3
6
9
12
-
```

2. Implement disjoint set operation

Program



```
File Edit Search Run Compile Debug Project Options Window Help
DISJOINT.C 2=1
#include<stdio.h>
#include<conio.h>
struct DisjSet
{
    int parent[10];
    int rank[10];
    int n;
}dis;

void makeSet()
{
    int i;
    for(i=0;i<dis.n;i++)
    {
        dis.parent[i]=i;
        dis.rank[i]=0;
    }
}

void displaySet()
{
    8:32
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

```
File Edit Search Run Compile Debug Project Options Window Help
DISJOINT.C 2=1
void displaySet()
{
    int i;
    printf("\n Parent array : \n");
    for(i=0;i<dis.n;i++)
    {
        printf("%d",dis.parent[i]);
    }
    printf("\n rank array : \n");
    for(i=0;i<dis.n;i++)
    {
        printf("%d",dis.rank[i]);
    }
    printf("\n");
}

int find(int x)
{
    if(dis.parent[x]!=x)
    {
        dis.parent[x]=find(dis.parent[x]);
    }
}

40:32
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

```
File Edit Search Run Compile Debug Project Options Window Help
DISJOINT.C 2=1
    }
    return dis.parent[x];
}

void Union(int x,int y)
{
    int xset=find(x);
    int yset=find(y);
    if(xset==yset)
        return;
    if(dis.rank[xset]<dis.rank[yset])
    {
        dis.parent[xset]=yset;
        dis.rank[xset]=-1;
    }
    else if(dis.rank[xset]>dis.rank[yset])
    {
        dis.parent[yset]=xset;
        dis.rank[yset]=-1;
    }
    else
    {
        dis.parent[xset]=yset;
        dis.rank[xset]=-1;
    }
}

61:32
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

```
File Edit Search Run Compile Debug Project Options Window Help
DISJOINT.C 2=[+]
{
    dis.parent[yset]=xset;
    dis.rank[xset]=dis.rank[xset]+1;
    dis.rank[yset]=-1;
}

int main()
{
    int n,x,y;
    int ch,wish;
    clrscr();
    printf("\nhow many elements ? ");
    scanf("%d",&dis.n);
    makeSet();

    printf("\n menu \n");
    printf("1.union,\n2.find,\n3.display\n");

    do
    {
        82:32
```

```
File Edit Search Run Compile Debug Project Options Window Help
DISJOINT.C 2=[+]
scanf("%d %d",&x,&y);
Union(x,y);
break;
case 2:
    printf("enter elements to check connected comp");
    scanf("%d %d",&x,&y);
    if (find(x)==find(y))
    {
        printf("\nconnected");
    }
    else
        printf("\n not connected");
    break;
case 3:displaySet();
    break;
}
printf("\ndo you wish to continue : (1/0)");
scanf("%d",&wish);
}while(wish==1);
return 0;
}
* 103:6
```

Output

how many elements ? 3

menu
1.union.
2.find.
3.display

Enter a choice : 1
enter the elements to union : 3
5

do you wish to ontinue : (1/0)1

Enter a choice : 2
enter elements to check connected component : 4
5

connected
do you wish to ontinue : (1/0)

1.union.
2.find.
3.display

Enter a choice : 1
enter the elements to union : 3
5

do you wish to ontinue : (1/0)1

Enter a choice : 2
enter elements to check connected component : 4
5

connected
do you wish to ontinue : (1/0)1

Enter a choice : 3

Parent array :
012
rank array :
000

do you wish to ontinue : (1/0)