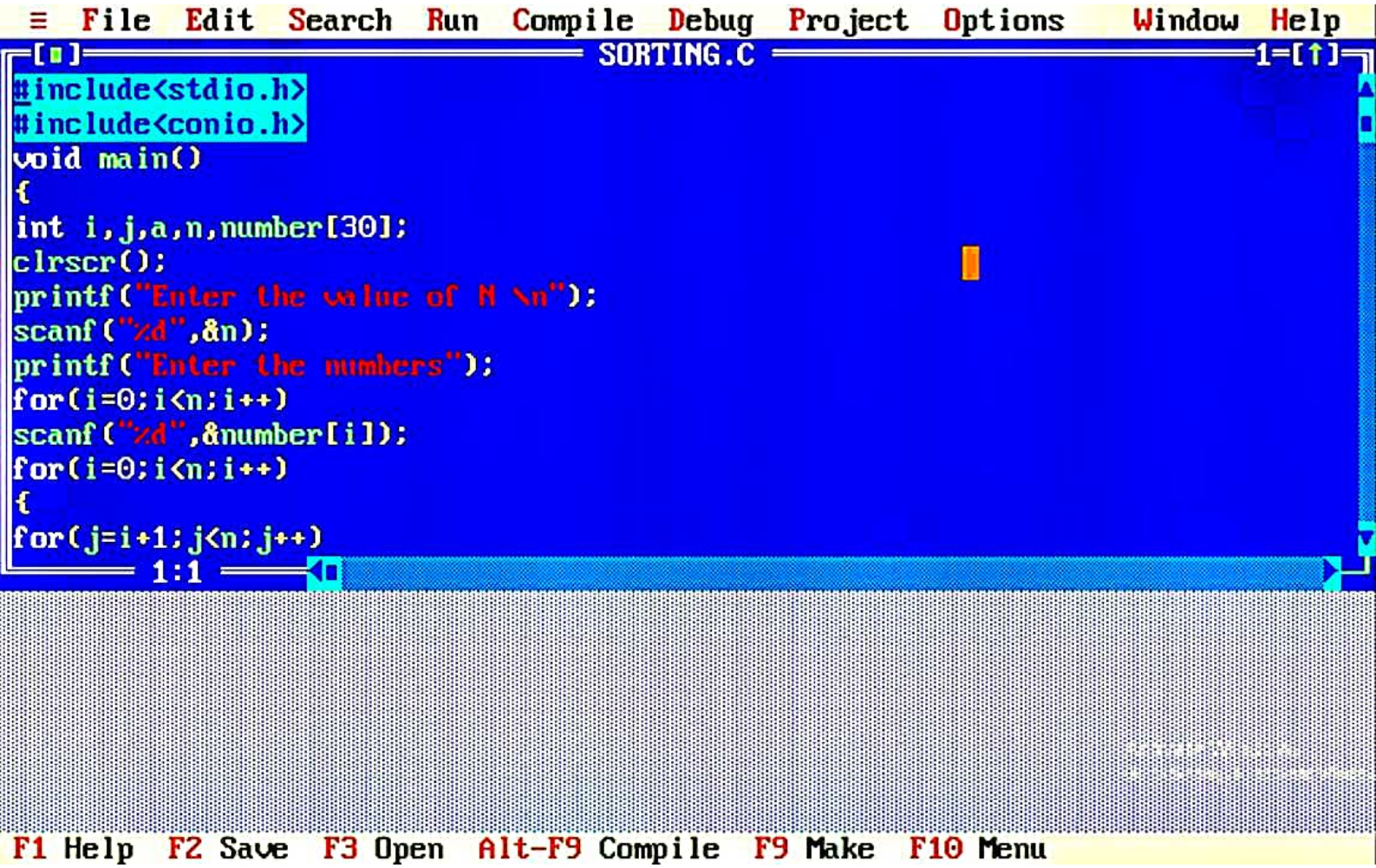
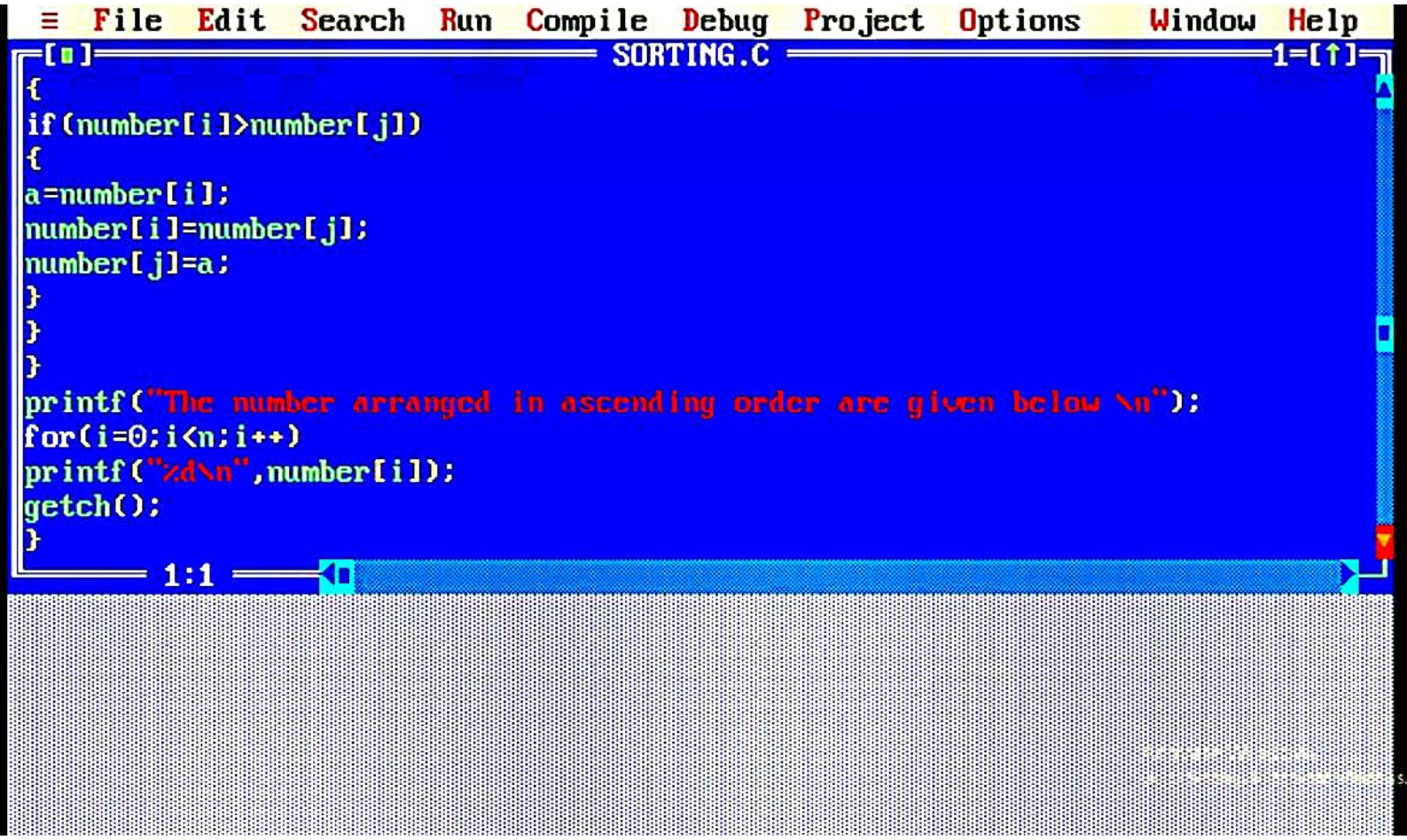
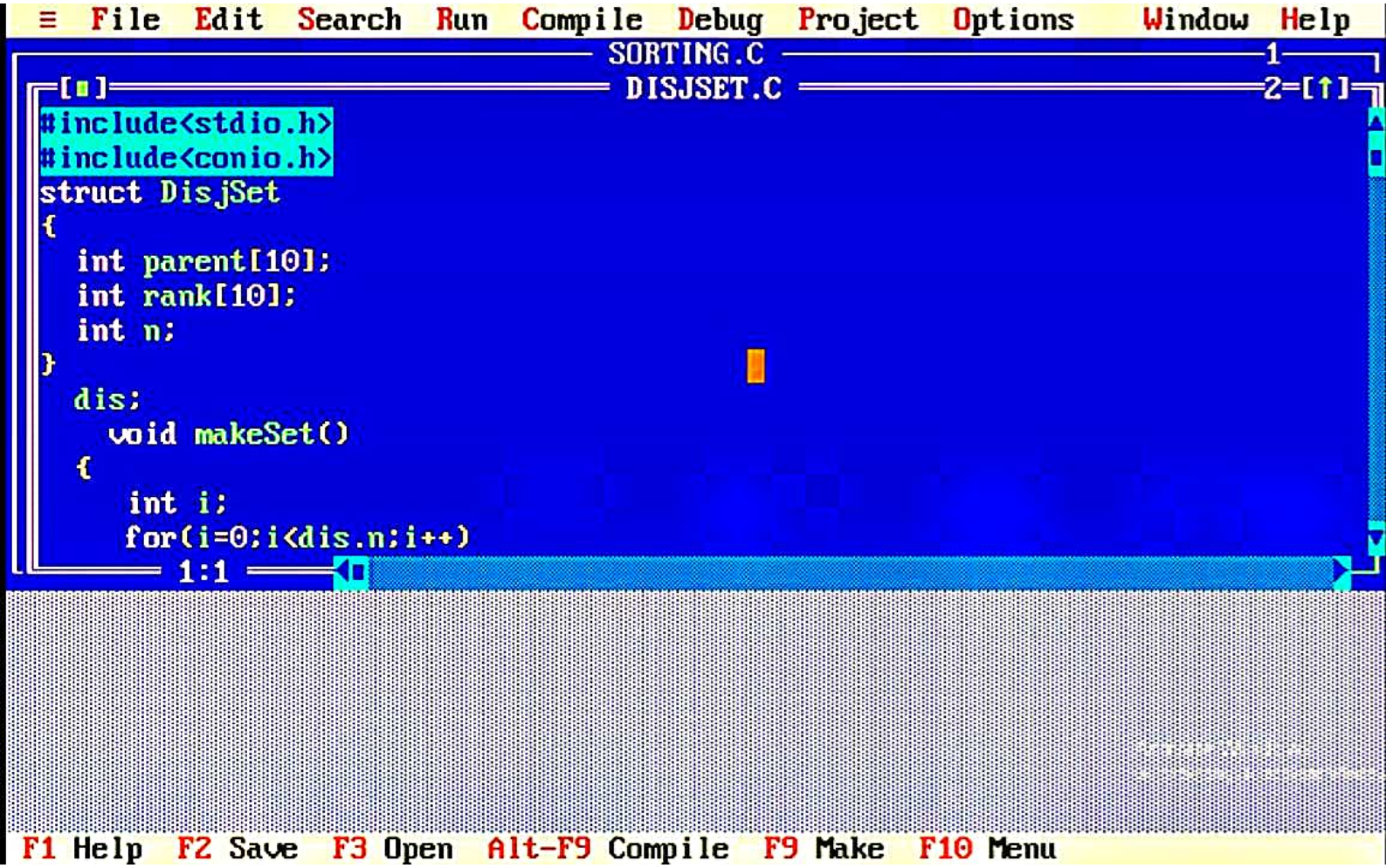
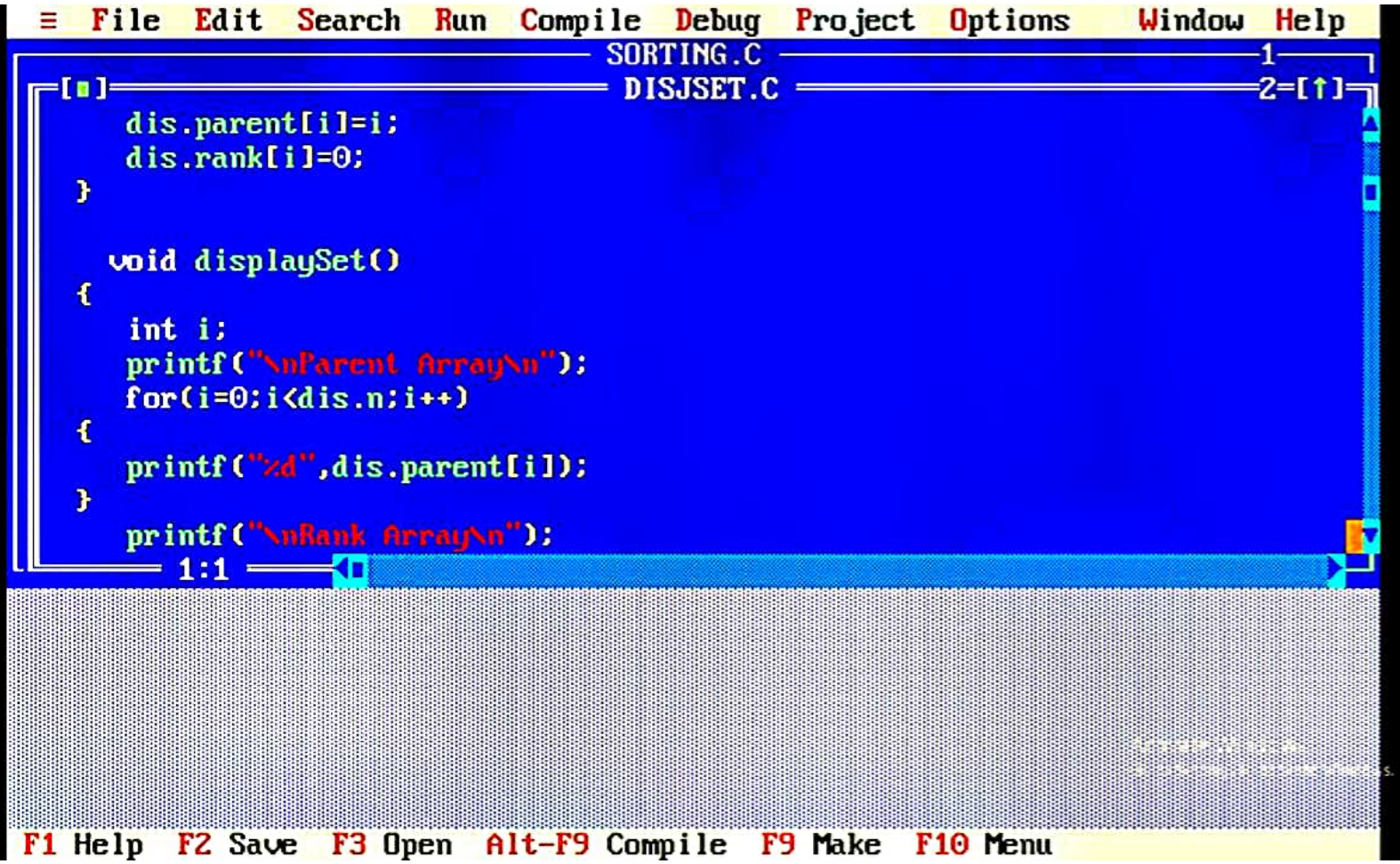
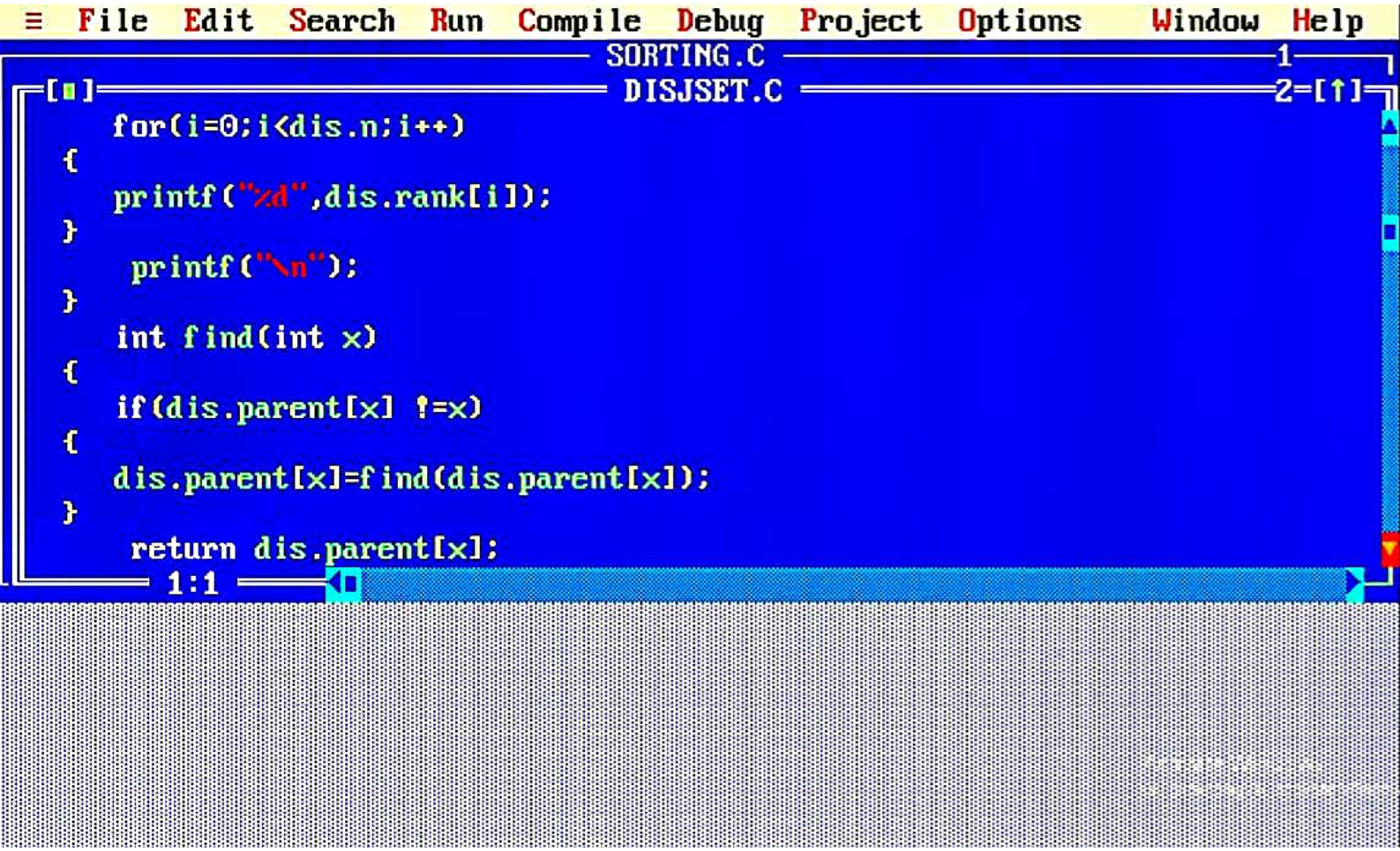

File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,a,n,number[30];
clrscr();
printf("Enter the value of N \n");
scanf("%d",&n);
printf("Enter the numbers");
for(i=0;i<n;i++)
scanf("%d",&number[i]);
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

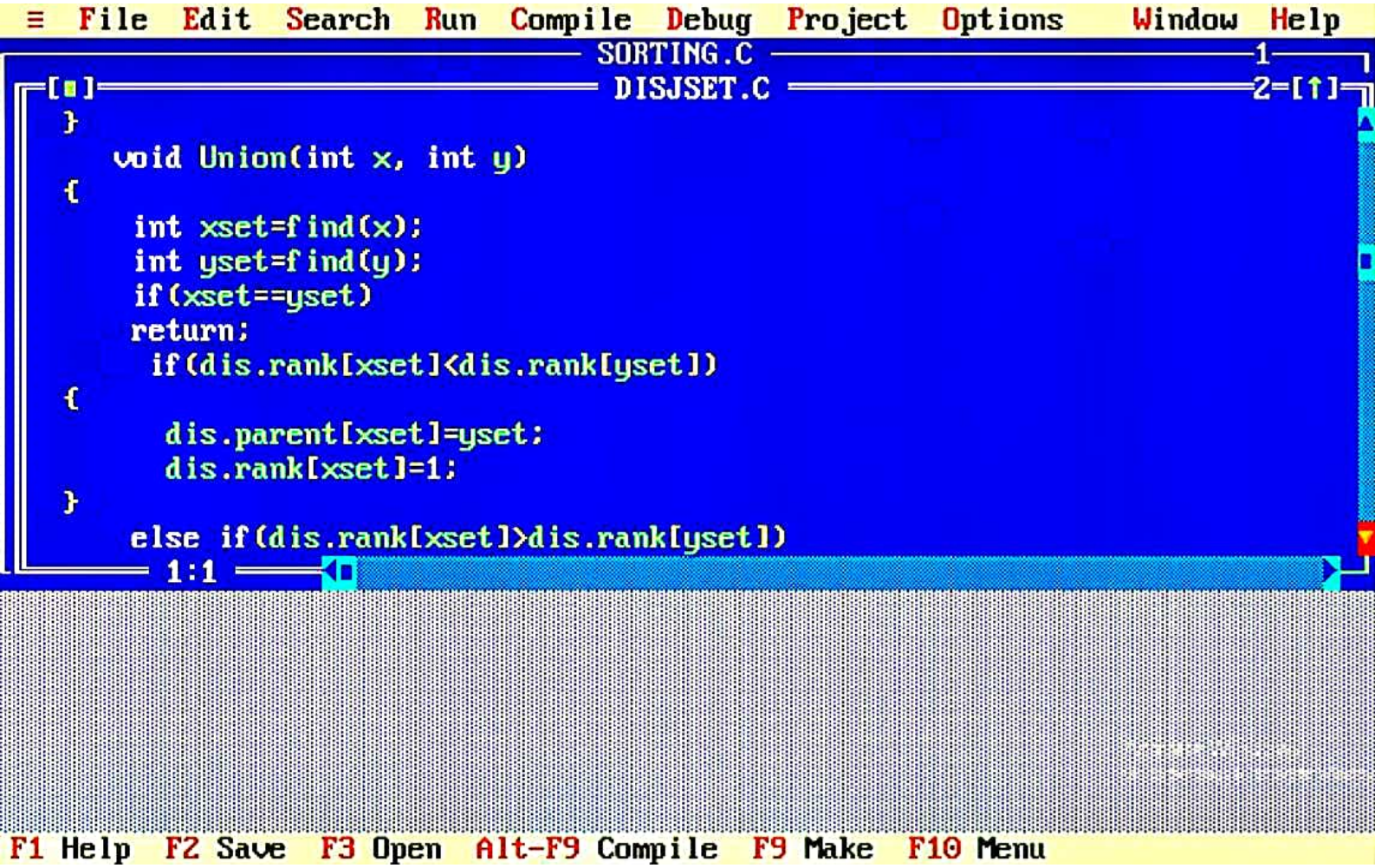

File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
{
if(number[i]>number[j])
{
a=number[i];
number[i]=number[j];
number[j]=a;
}
}
}
printf("The number arranged in ascending order are given below \n");
for(i=0;i<n;i++)
printf("%d\n",number[i]);
getch();
}
1:1

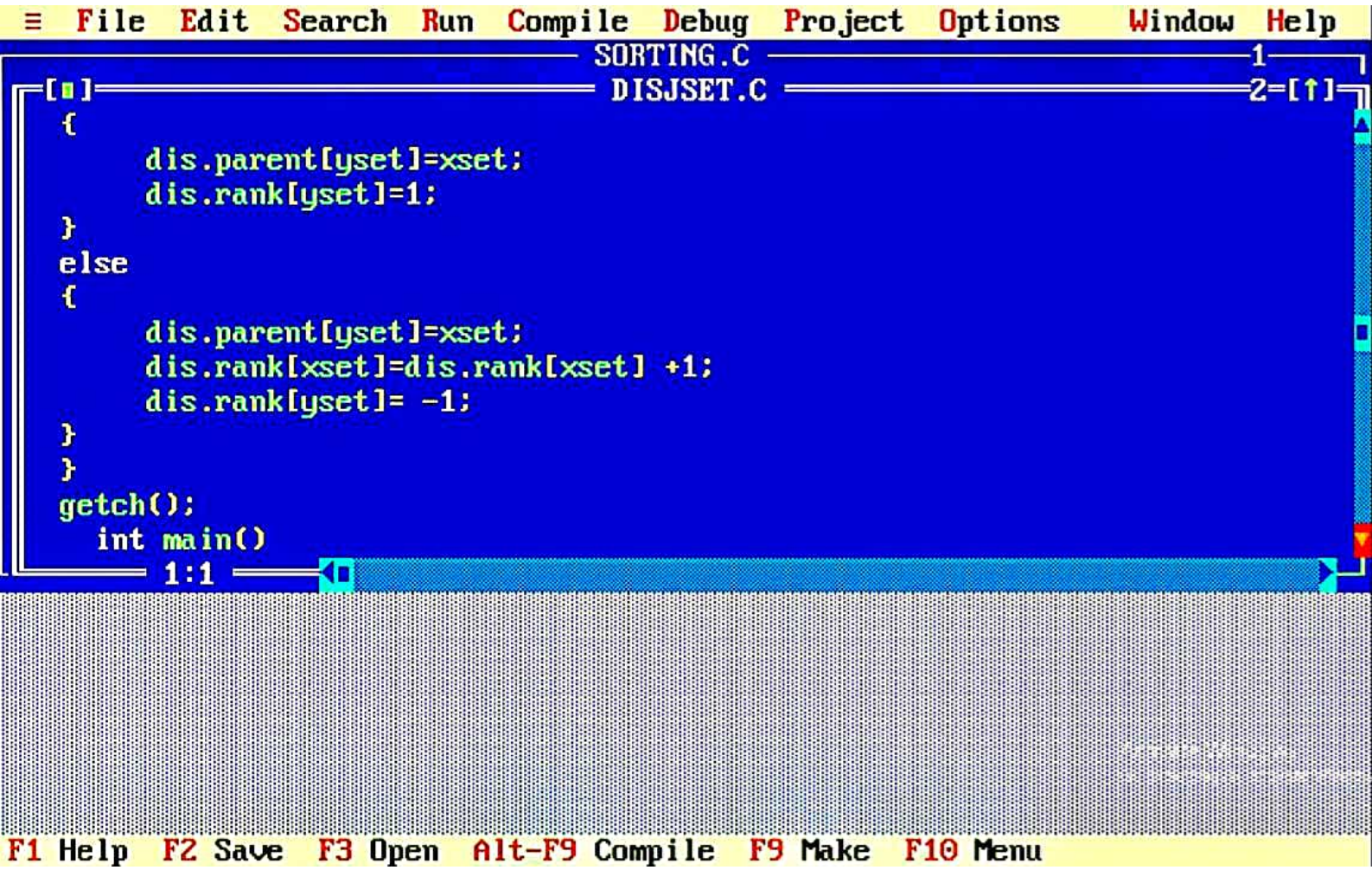
```
Enter the value of N
5
Enter the numbers 1 8 5 2 4
The number arranged in ascending order are given below
1
2
4
5
8
```

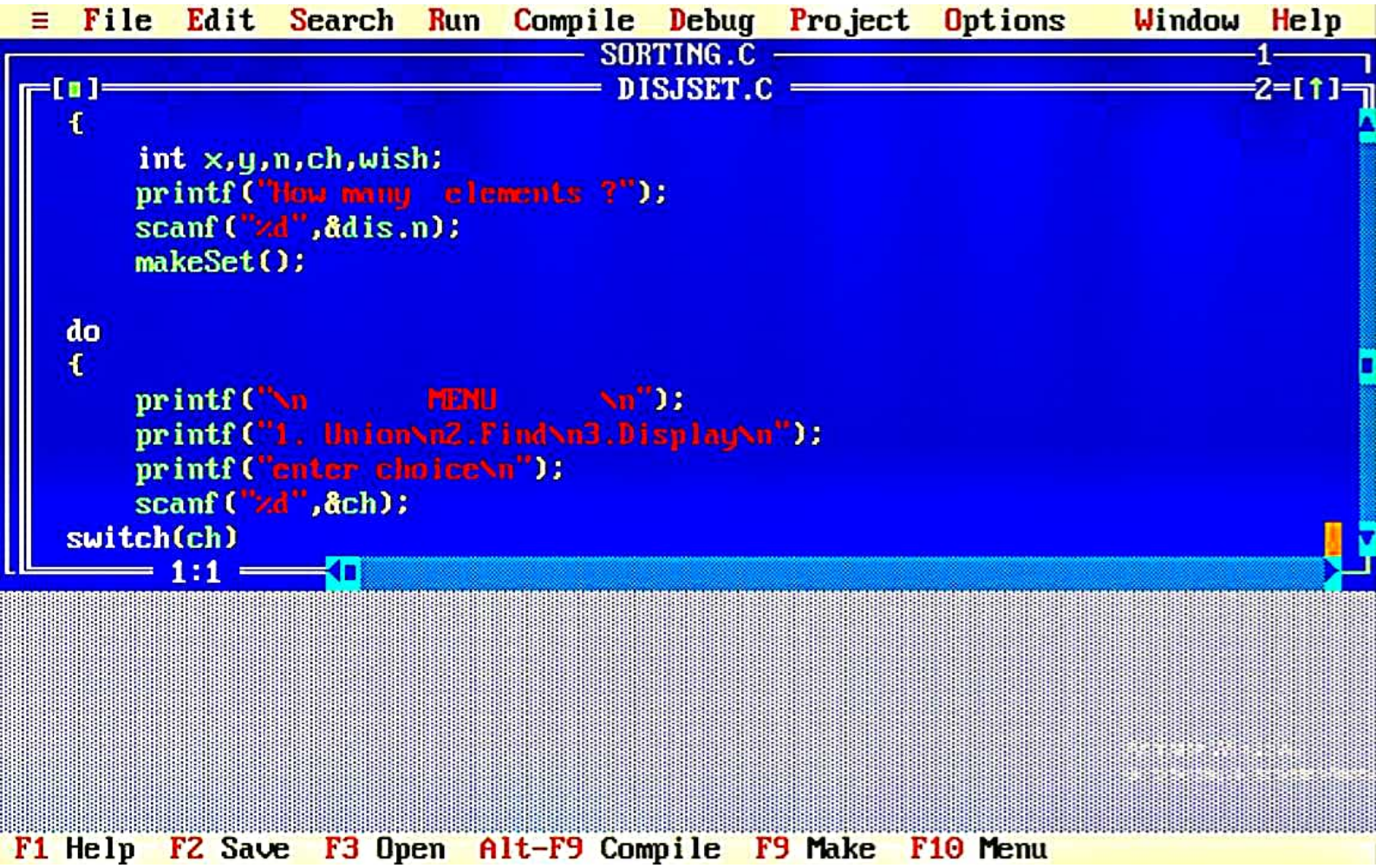

File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
#include<stdio.h>
#include<conio.h>
struct DisjSet
{
int parent[10];
int rank[10];
int n;
}
dis;
void makeSet()
{
int i;
for(i=0;i<dis.n;i++)
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

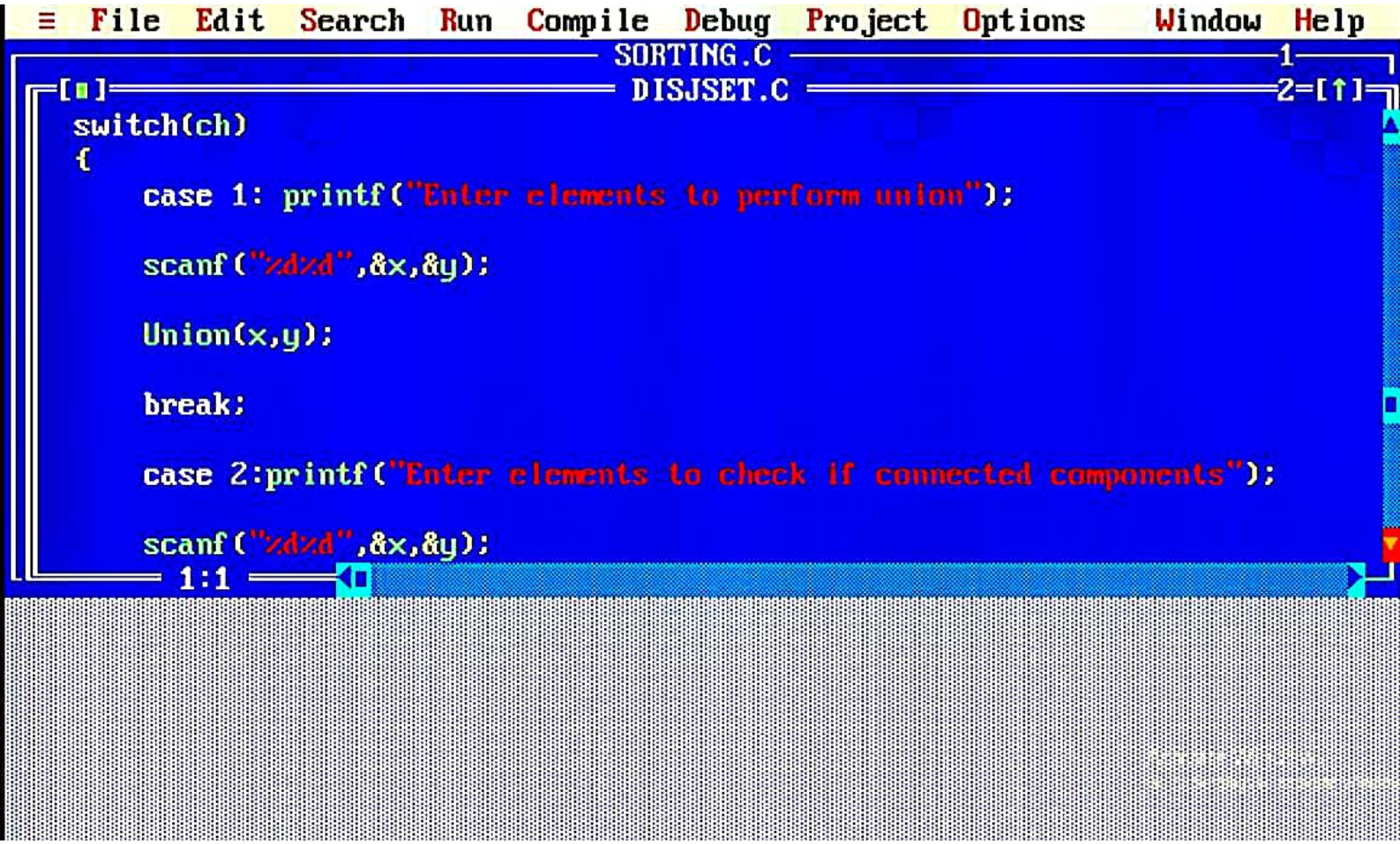

File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
dis.parent[i]=i;
dis.rank[i]=0;
}
void displaySet()
{
int i;
printf("\nParent Array\n");
for(i=0;i<dis.n;i++)
{
printf("%d",dis.parent[i]);
}
printf("\nRank Array\n");
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

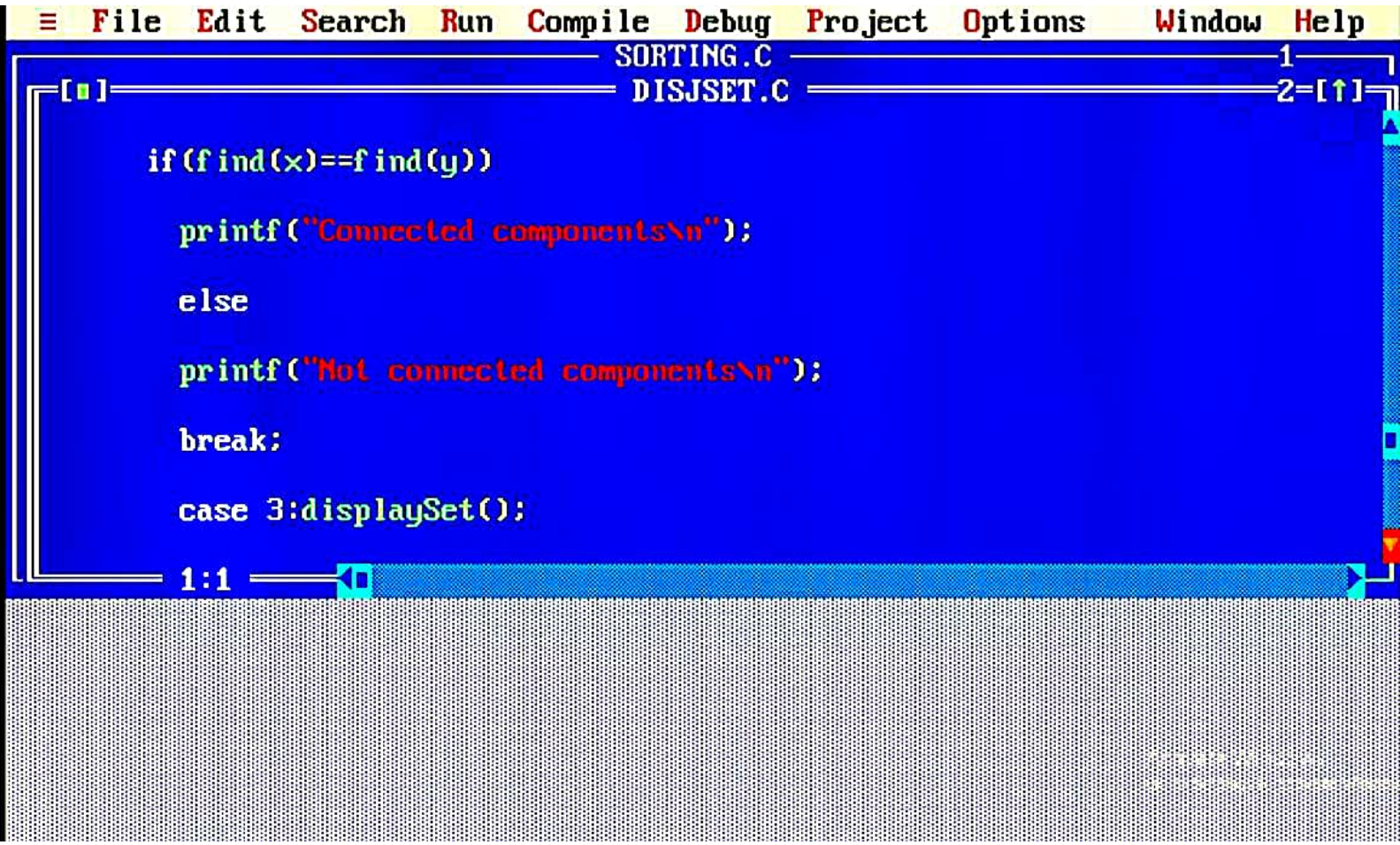

File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
for(i=0;i<dis.n;i++)
{
printf("%d",dis.rank[i]);
}
printf("\n");
}
int find(int x)
{
if(dis.parent[x] !=x)
{
dis.parent[x]=find(dis.parent[x]);
}
return dis.parent[x];
1:1

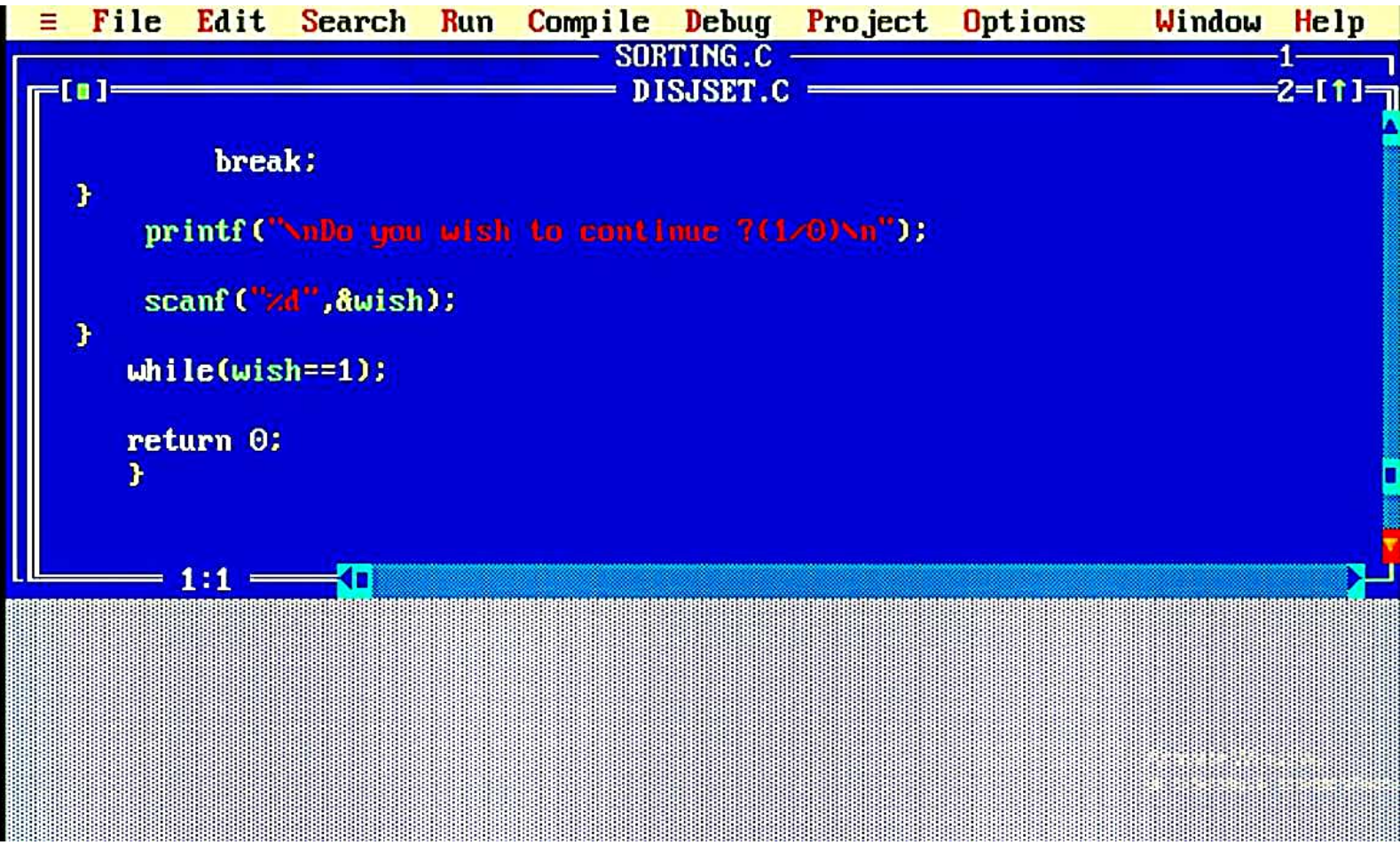

File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
}
void Union(int x, int y)
{
int xset=find(x);
int yset=find(y);
if(xset==yset)
return;
if(dis.rank[xset]<dis.rank[yset])
{
dis.parent[xset]=yset;
dis.rank[xset]=1;
}
else if(dis.rank[xset]>dis.rank[yset])
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
{
dis.parent[yset]=xset;
dis.rank[yset]=1;
}
else
{
dis.parent[yset]=xset;
dis.rank[xset]=dis.rank[xset] +1;
dis.rank[yset]= -1;
}
}
getch();
int main()
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
SORTING.C
DISJSET.C
{
int x,y,n,ch,wish;
printf("How many elements ?");
scanf("%d",&dis.n);
makeSet();
do
{
printf("\n MENU \n");
printf("1. Union\n2.Find\n3.Display\n");
printf("enter choice\n");
scanf("%d",&ch);
switch(ch)
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
SORTING.C
DISJSET.C
switch(ch)
{
case 1: printf("Enter elements to perform union");
scanf("%d%d",&x,&y);
Union(x,y);
break;
case 2:printf("Enter elements to check if connected components");
scanf("%d%d",&x,&y);
1:1


File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
if(find(x)==find(y))
printf("Connected components\n");
else
printf("Not connected components\n");
break;
case 3:displaySet();
1:1


File Edit Search Run Compile Debug Project Options Window Help
SORTING.C
DISJSET.C
break;
}
printf("\nDo you wish to continue ?(1/0)\n");
scanf("%d",&wish);
}
while(wish==1);
return 0;
}
1:1