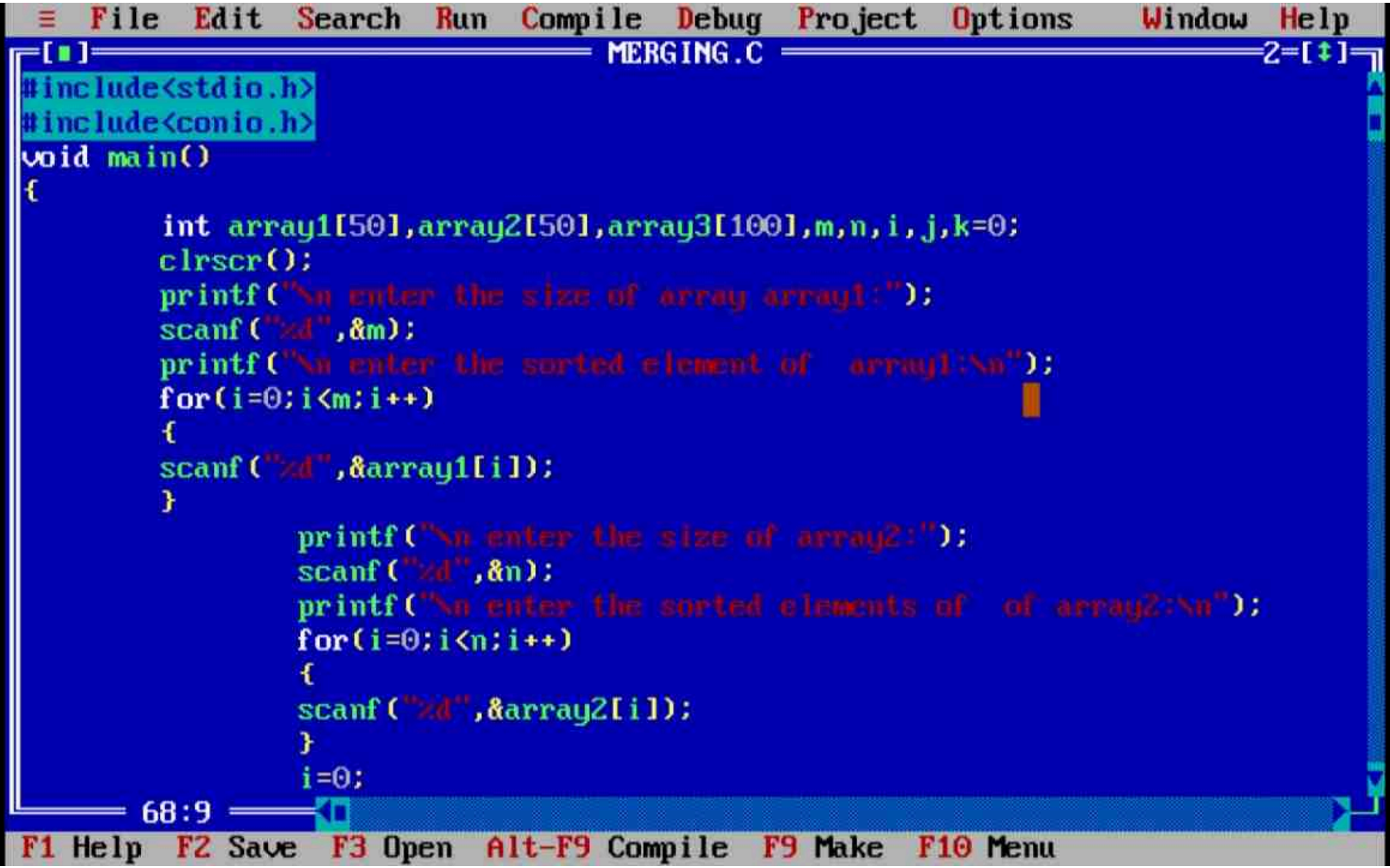
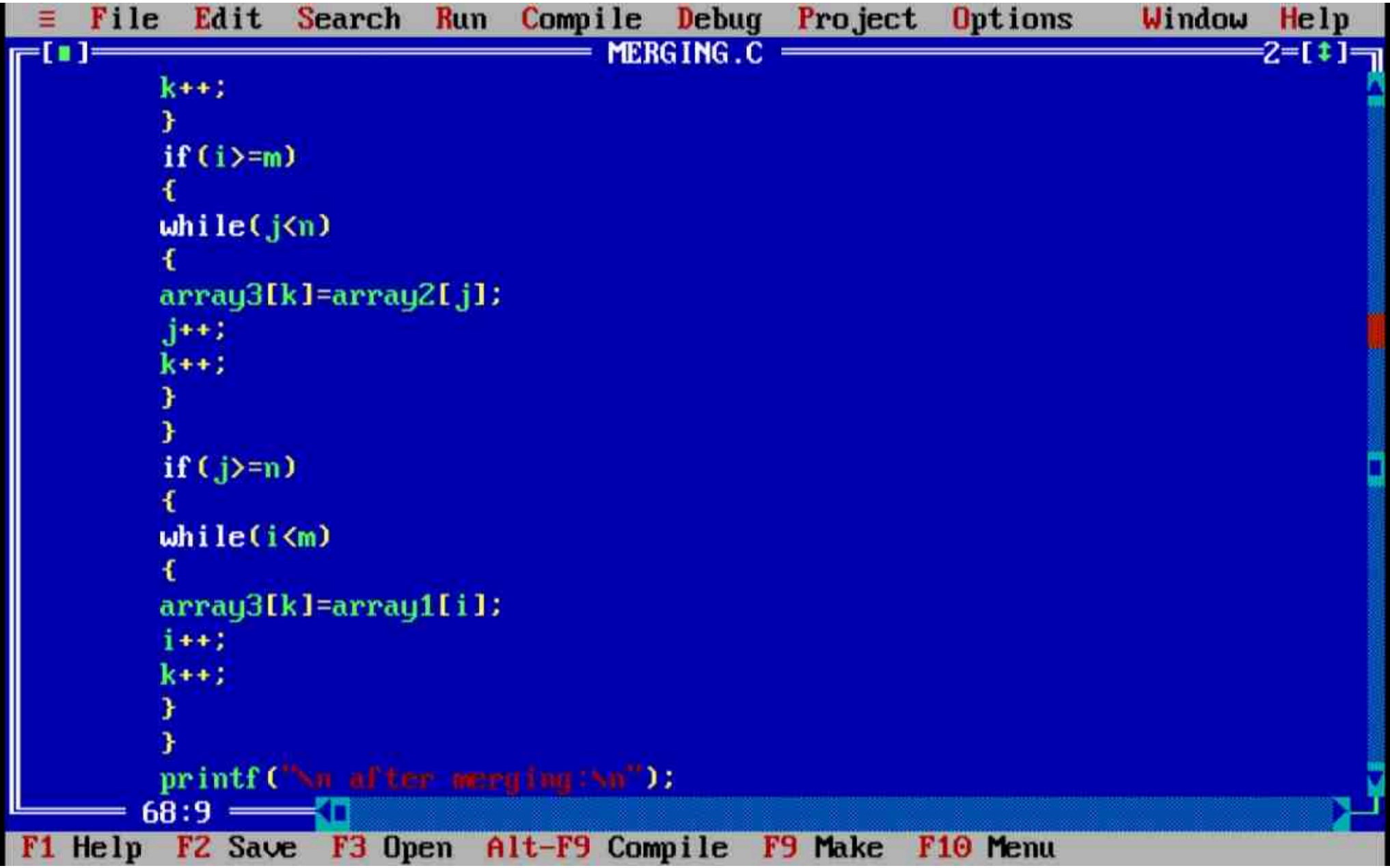
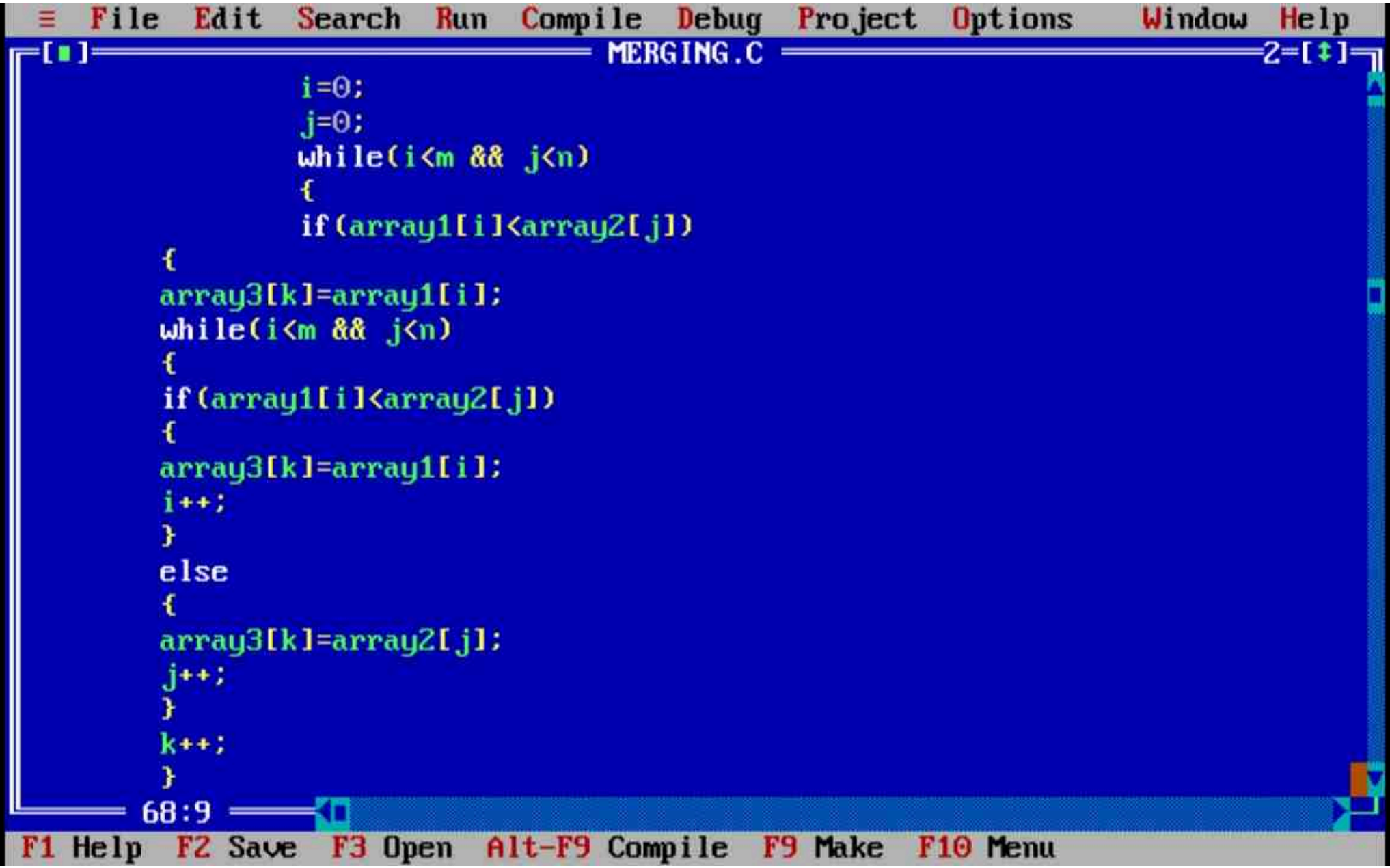
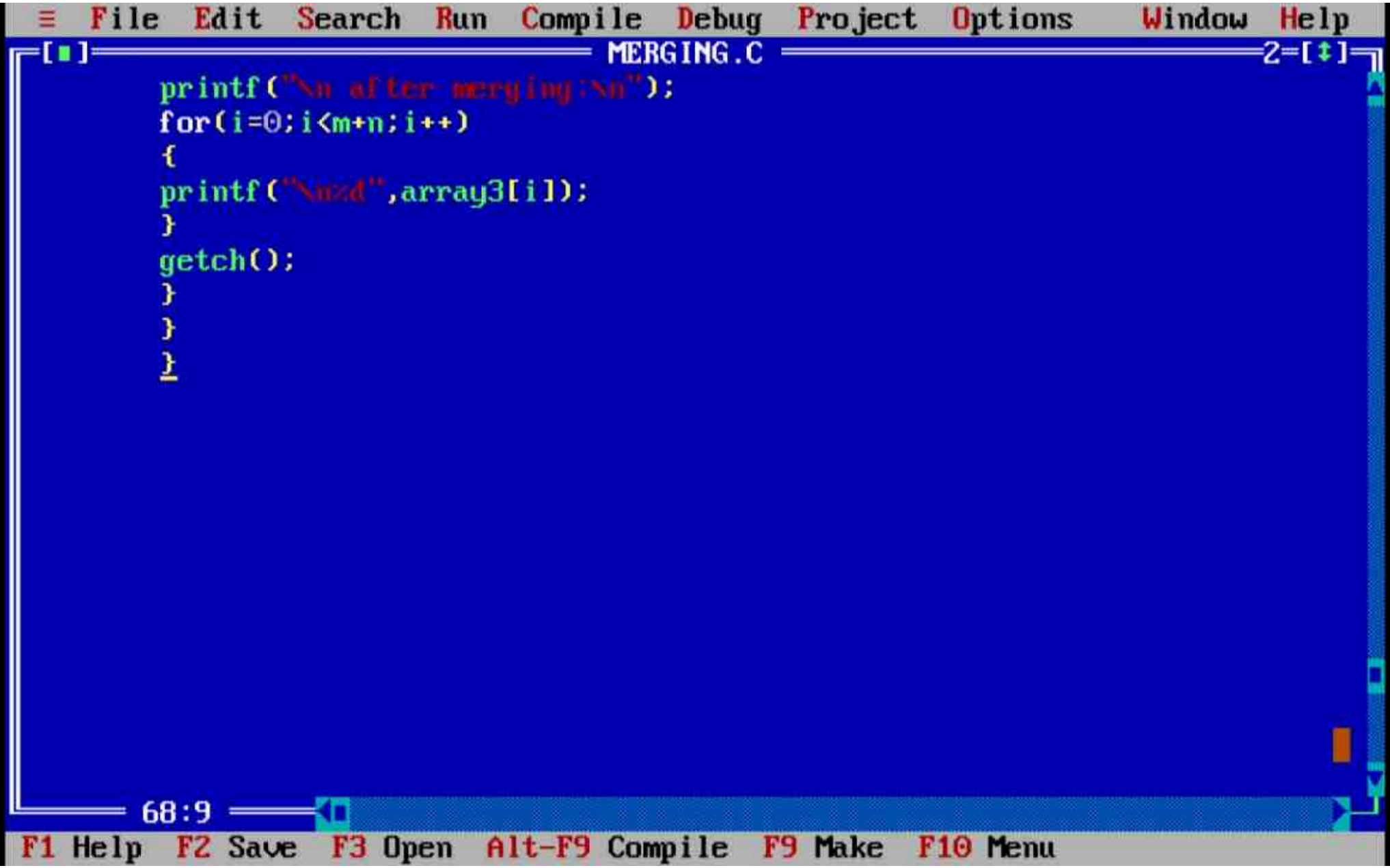
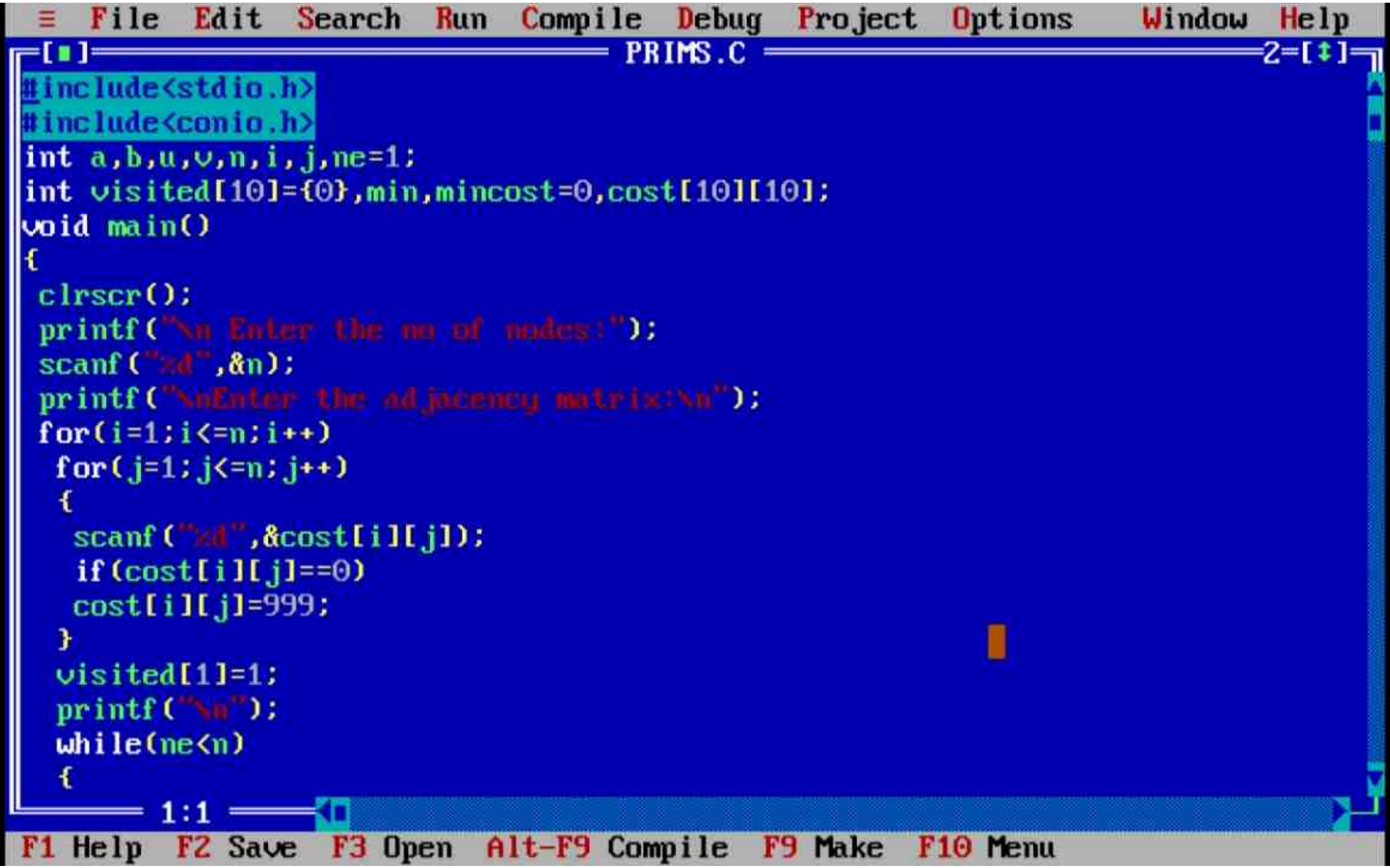

File Edit Search Run Compile Debug Project Options Window Help
MERGING.C
#include<stdio.h>
#include<conio.h>
void main()
{
int array1[50],array2[50],array3[100],m,n,i,j,k=0;
clrscr();
printf("\n enter the size of array array1:");
scanf("%d",&m);
printf("\n enter the sorted element of array1:\n");
for(i=0;i<m;i++)
{
scanf("%d",&array1[i]);
}
printf("\n enter the size of array2:");
scanf("%d",&n);
printf("\n enter the sorted elements of of array2:\n");
for(i=0;i<n;i++)
{
scanf("%d",&array2[i]);
}
i=0;
68:9
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

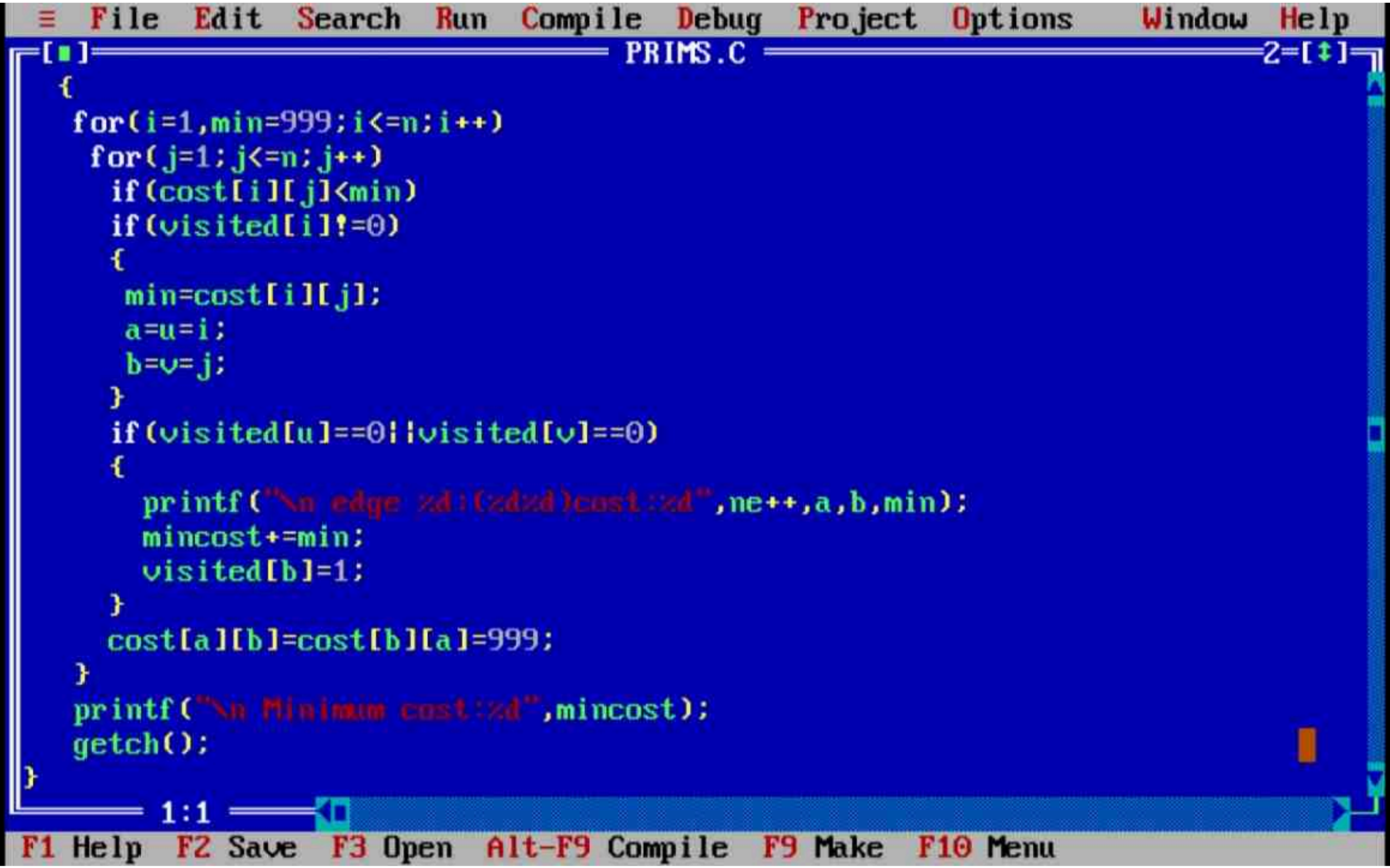

File Edit Search Run Compile Debug Project Options Window Help
MERGING.C
k++;
}
if(i>=m)
{
while(j<n)
{
array3[k]=array2[j];
j++;
k++;
}
}
if(j>=n)
{
while(i<m)
{
array3[k]=array1[i];
i++;
k++;
}
}
printf("\n after merging:\n");
68:9
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
MERGING.C
i=0;
j=0;
while(i<m && j<n)
{
if(array1[i]<array2[j])
{
array3[k]=array1[i];
while(i<m && j<n)
{
if(array1[i]<array2[j])
{
array3[k]=array1[i];
i++;
}
else
{
array3[k]=array2[j];
j++;
}
k++;
}
68:9
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
MERGING.C
printf("\n after meryimg:\n");
for(i=0;i<m+n;i++)
{
printf("\n%d",array3[i]);
}
getch();
}
}
}
68:9
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
 enter the size of array array1:3

 enter the sorted element of  array1:
1 2 3

 enter the size of array2:3

 enter the sorted elements of  of array2:
4 5 6

 after merging:

1
2
3
4
5
6
```


File Edit Search Run Compile Debug Project Options Window Help
PRIMS.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("\n Enter 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
PRIMS.C
{
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 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_
```