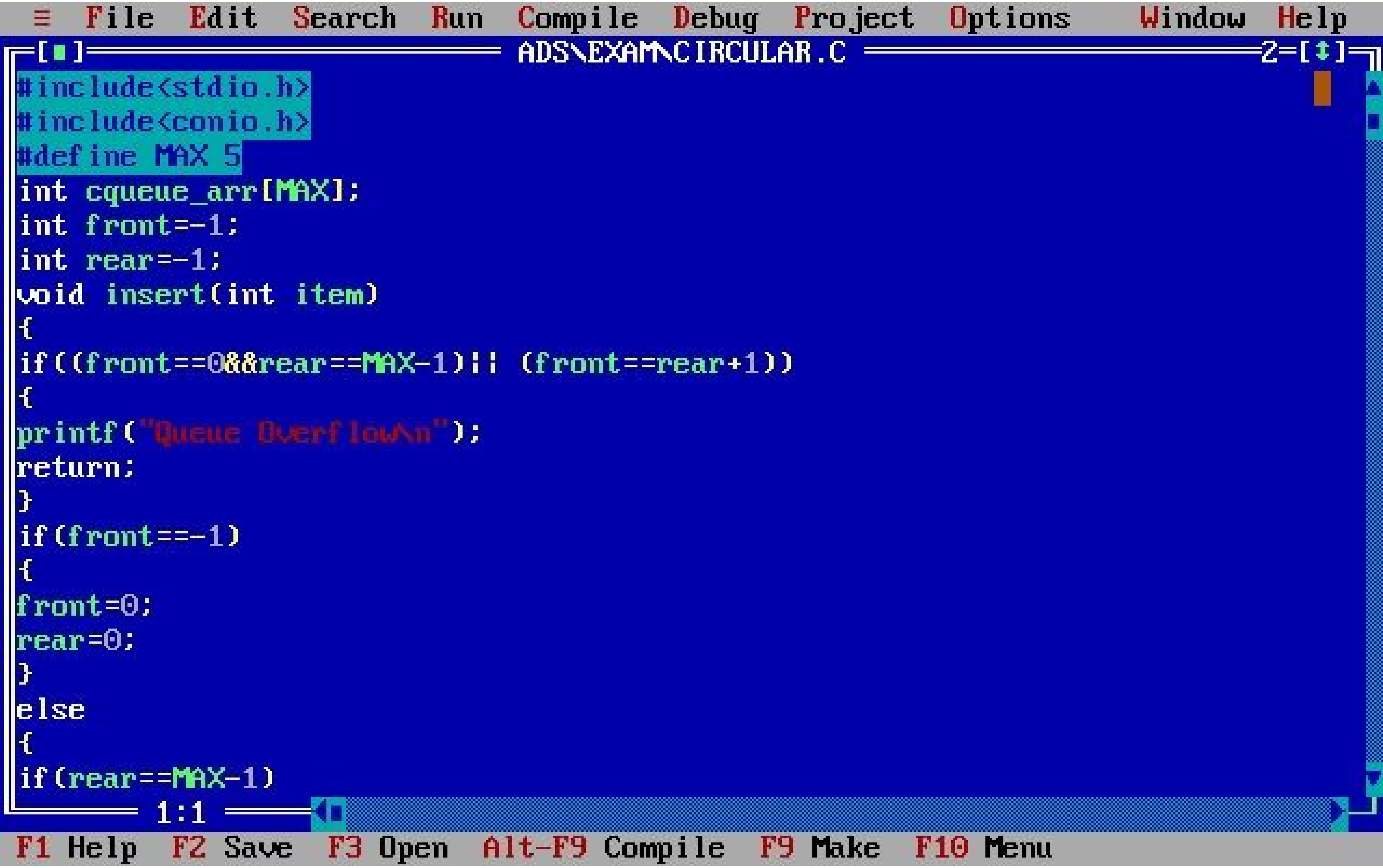
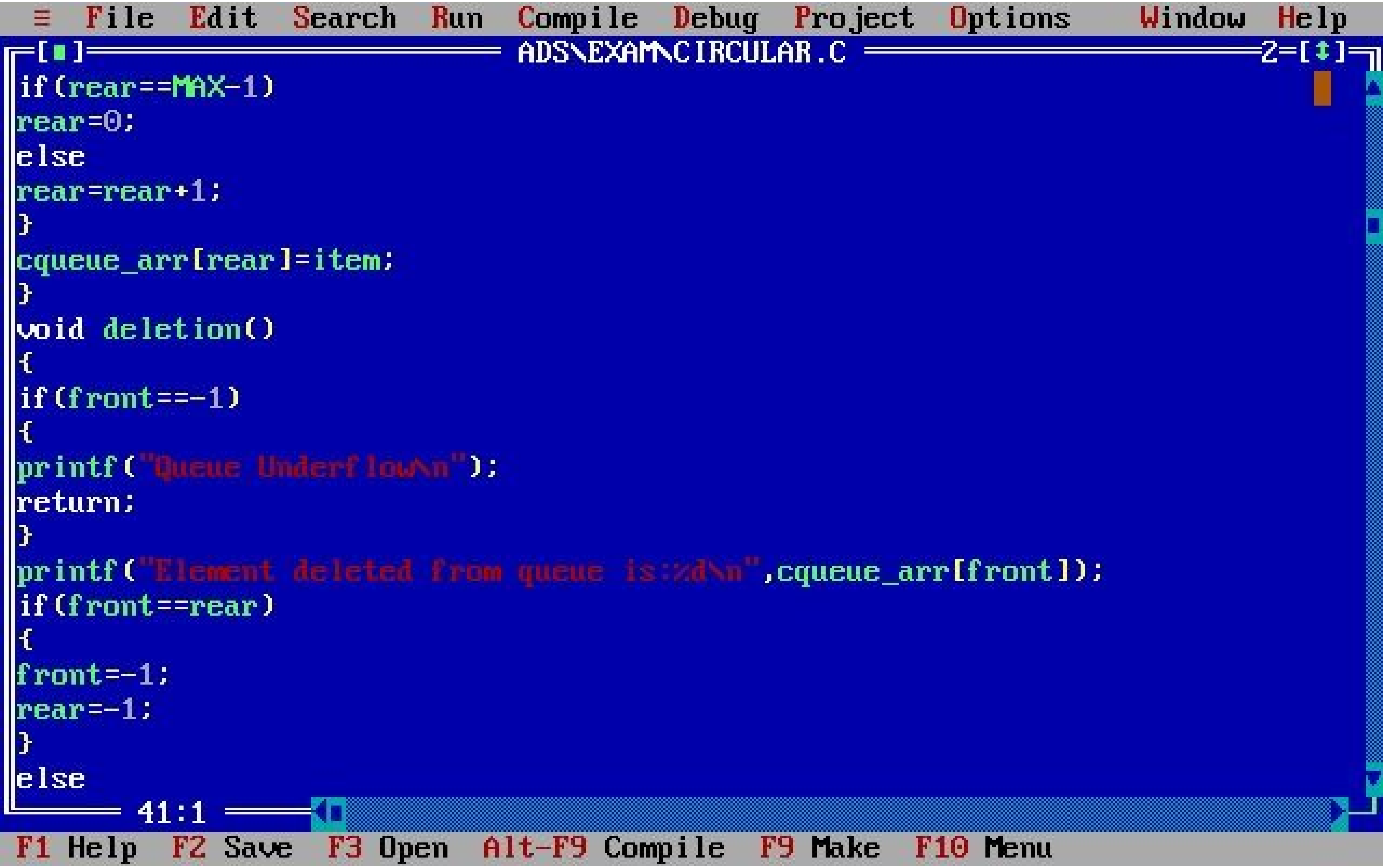
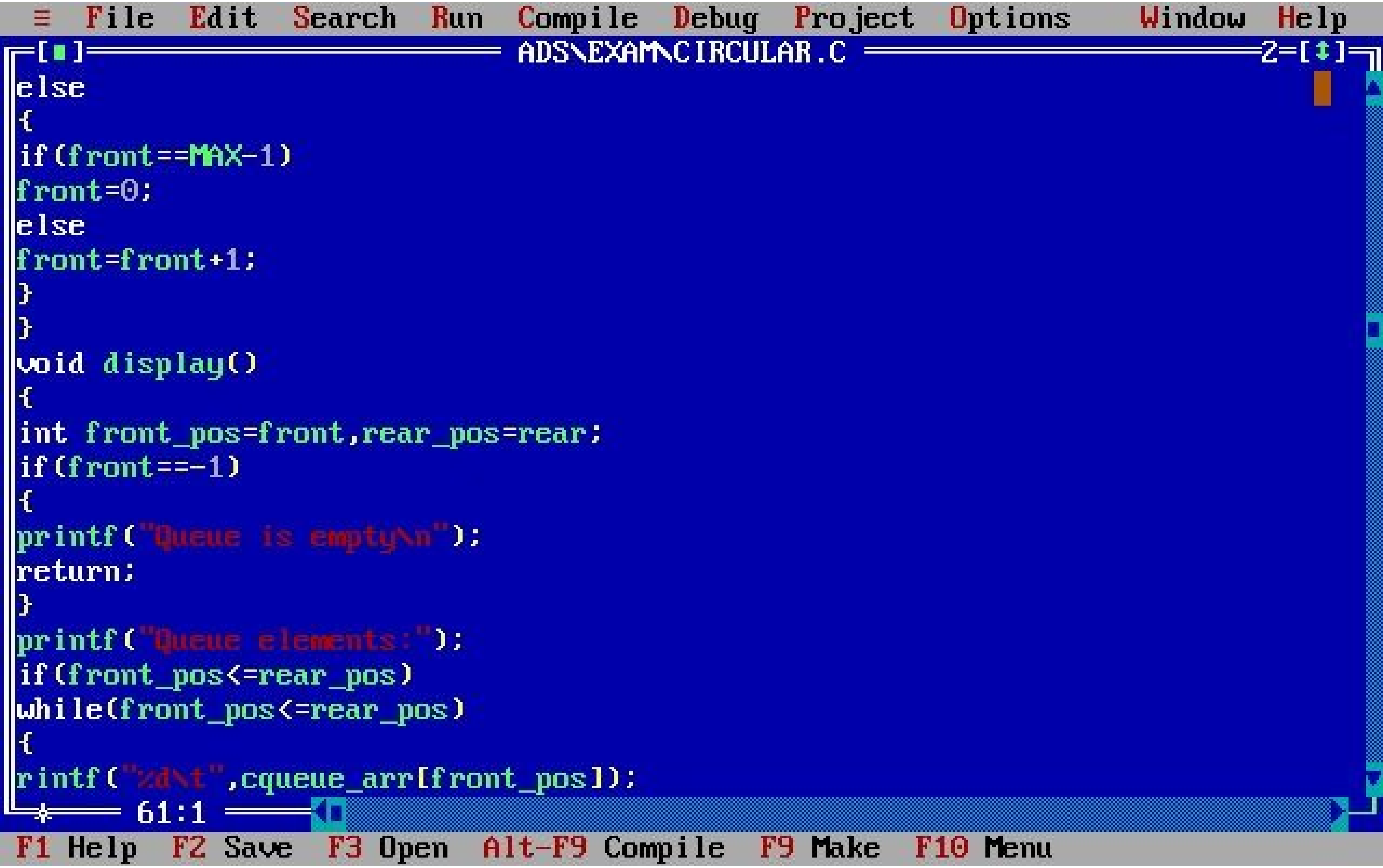

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] ADS\EXAM\CIRCULAR.C 2=[↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

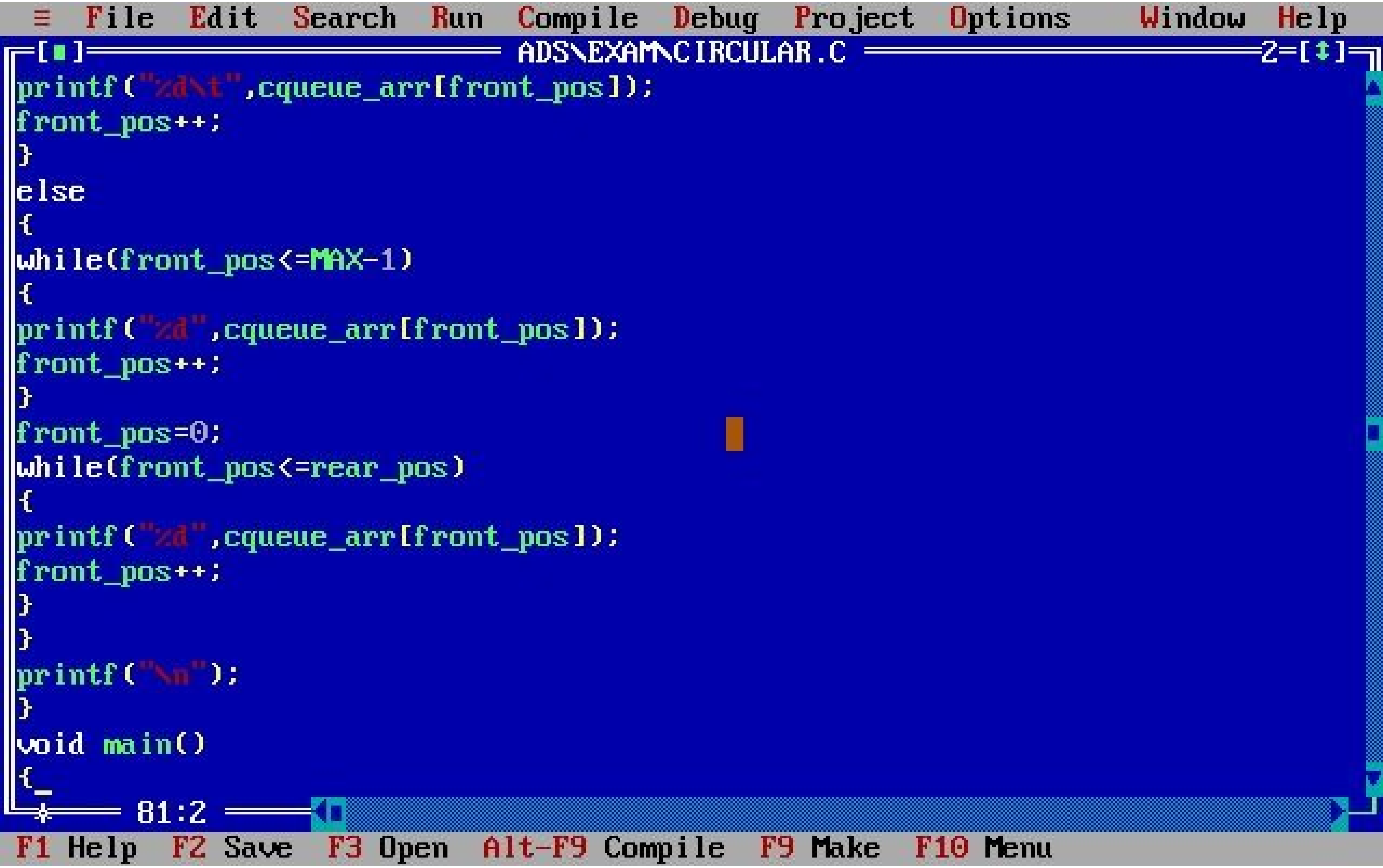
```c
#include<stdio.h>
#include<conio.h>
#define MAX 5
int cqueue_arr[MAX];
int front=-1;
int rear=-1;
void insert(int item)
{
if((front==0&&rear==MAX-1)|| (front==rear+1))
{
printf("Queue Overflow\n");
return;
}
if(front==-1)
{
front=0;
rear=0;
}
else
{
if(rear==MAX-1)
```


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] ADS\EXAM\CIRCULAR.C 2=[↕]
41:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

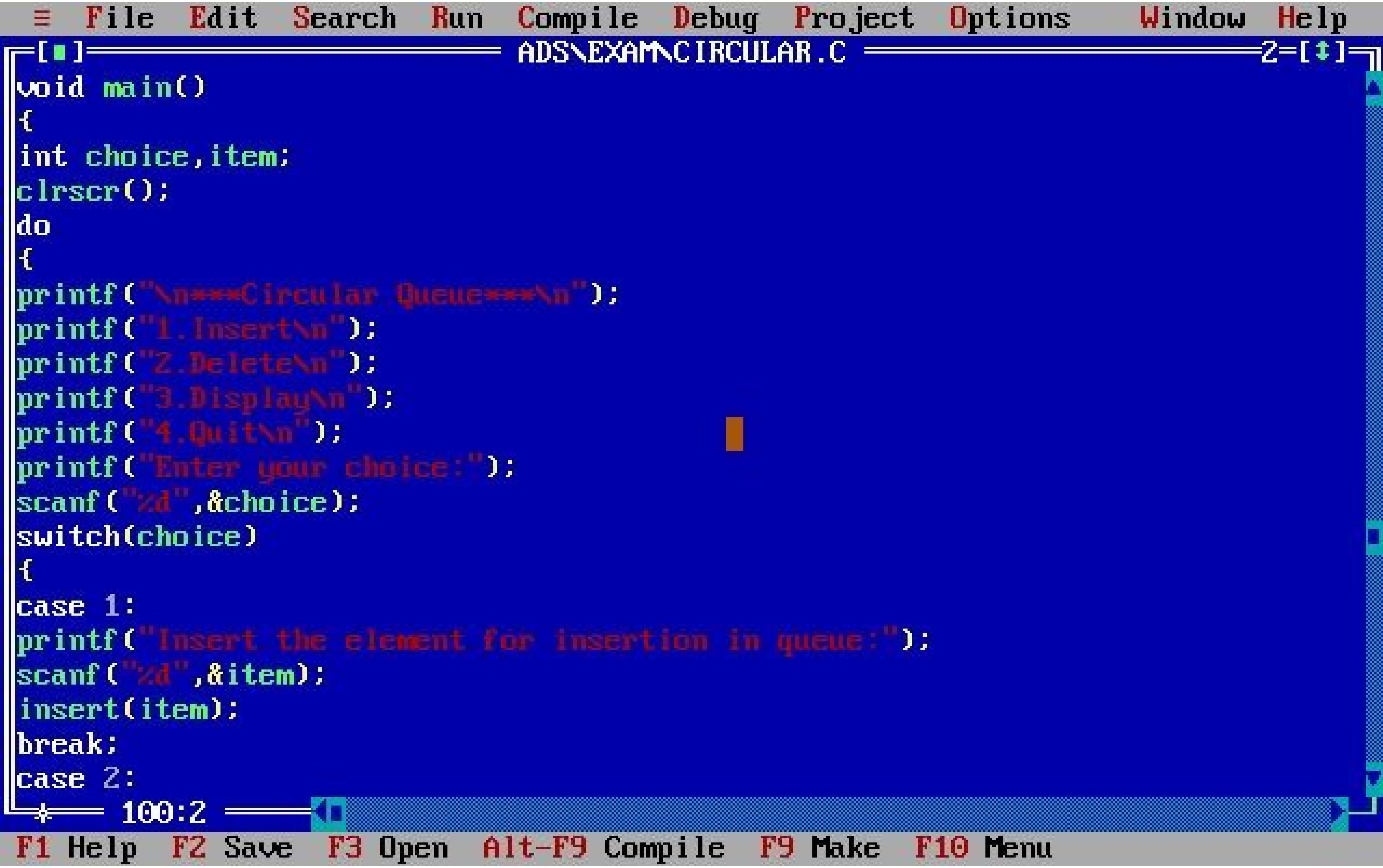
```
if(rear==MAX-1)
rear=0;
else
rear=rear+1;
}
cqueue_arr[rear]=item;
}
void deletion()
{
if(front==-1)
{
printf("Queue Underflow\n");
return;
}
printf("Element deleted from queue is:%d\n",cqueue_arr[front]);
if(front==rear)
{
front=-1;
rear=-1;
}
else
```


≡ File Edit Search Run Compile Debug Project Options Window Help
ADS\EXAM\CIRCULAR.C
2
61:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

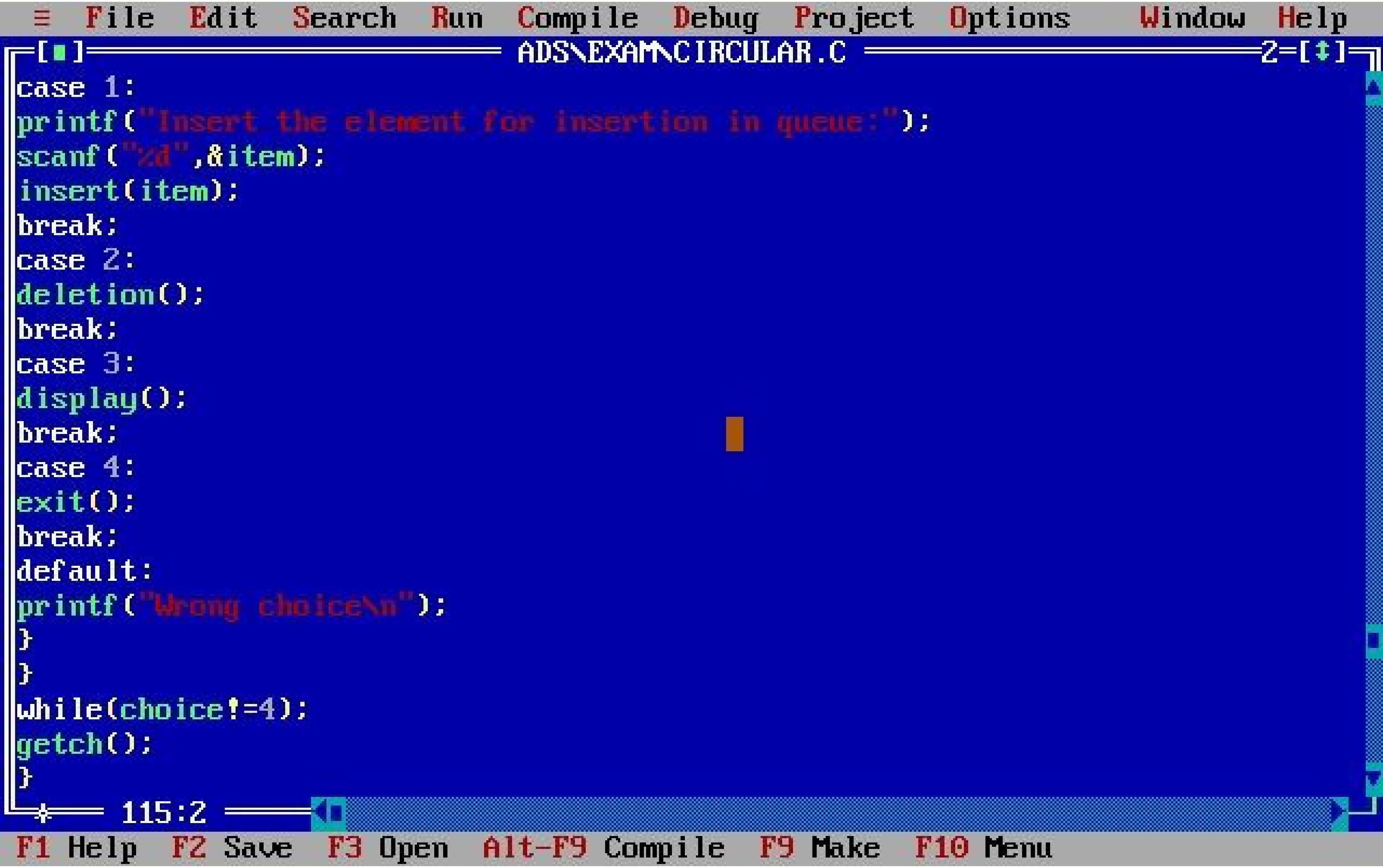
```
else
{
if(front==MAX-1)
front=0;
else
front=front+1;
}
}
void display()
{
int front_pos=front,rear_pos=rear;
if(front==-1)
{
printf("Queue is empty\n");
return;
}
printf("Queue elements:");
if(front_pos<=rear_pos)
while(front_pos<=rear_pos)
{
rintf("%d\t",cqueue_arr[front_pos]);
```


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] ADS\EXAM\CIRCULAR.C 2=[↕]
81:2
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
printf("%d\t",cqueue_arr[front_pos]);
front_pos++;
}
else
{
while(front_pos<=MAX-1)
{
printf("%d",cqueue_arr[front_pos]);
front_pos++;
}
front_pos=0;
while(front_pos<=rear_pos)
{
printf("%d",cqueue_arr[front_pos]);
front_pos++;
}
}
printf("\n");
}
void main()
{_
```

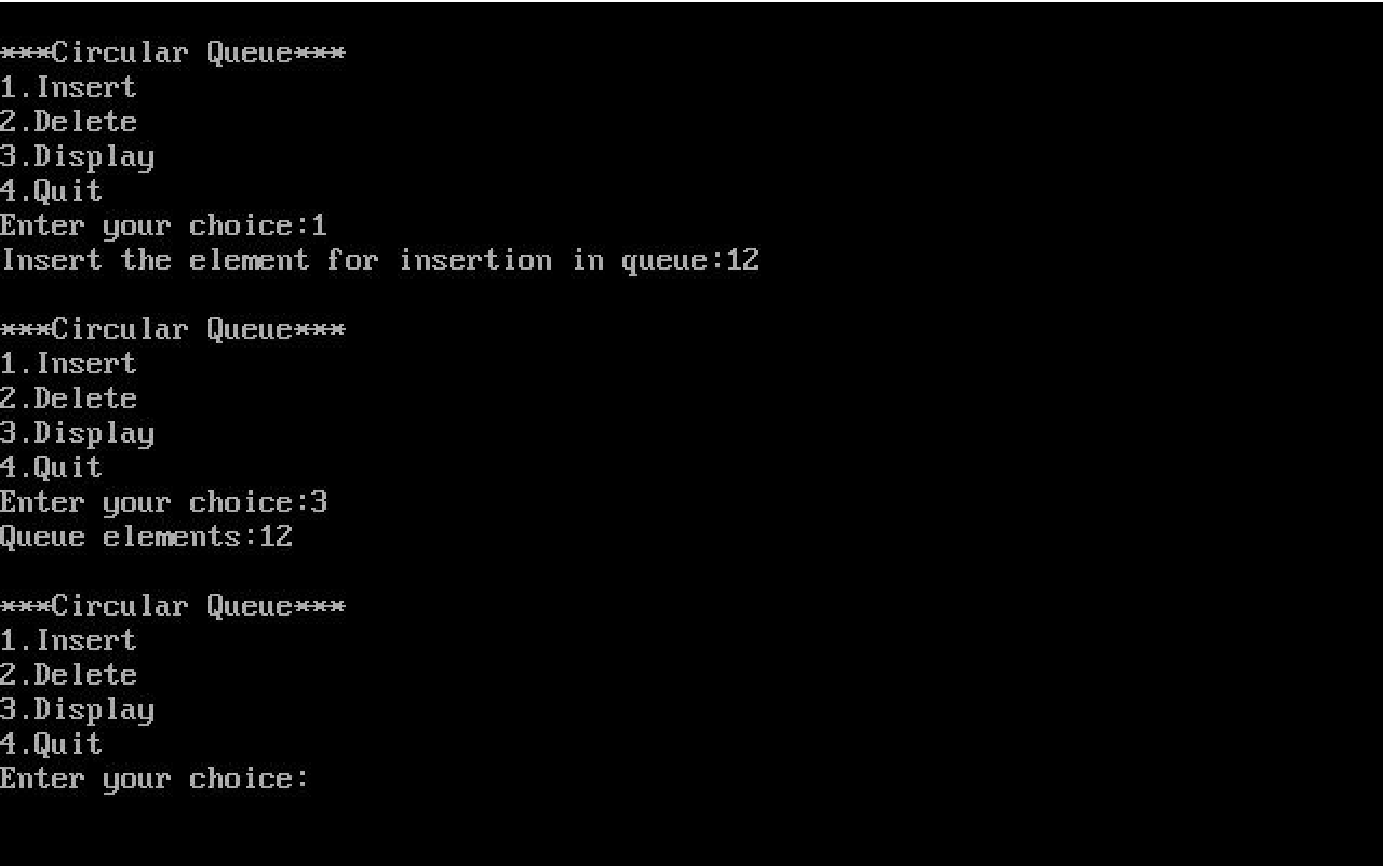

≡ File Edit Search Run Compile Debug Project Options Window Help
ADS\EXAM\CIRCULAR.C
2
100:2
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
void main()
{
int choice,item;
clrscr();
do
{
printf("\n***Circular Queue***\n");
printf("1.Insert\n");
printf("2.Delete\n");
printf("3.Display\n");
printf("4.Quit\n");
printf("Enter your choice:");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Insert the element for insertion in queue:");
scanf("%d",&item);
insert(item);
break;
case 2:
```

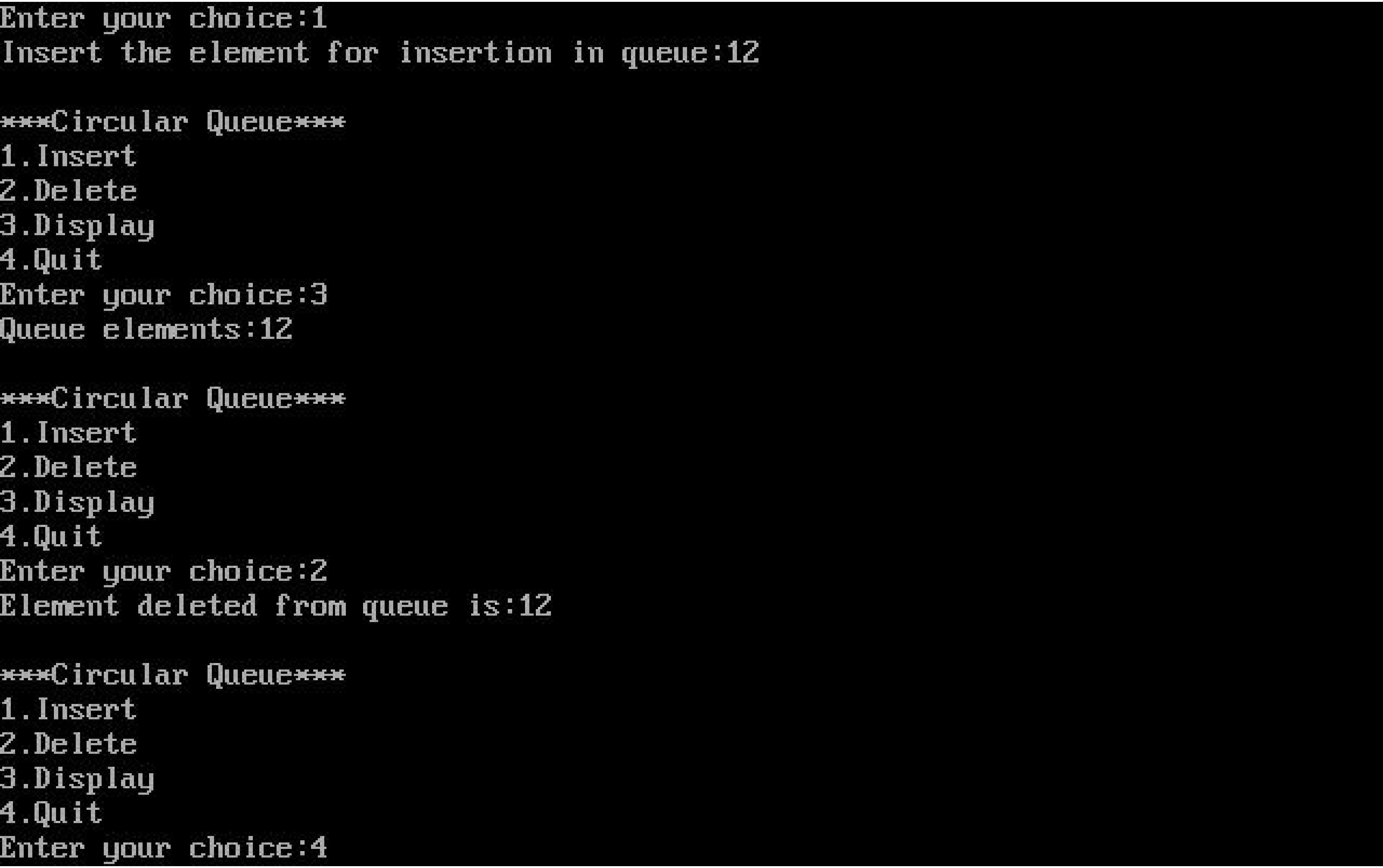


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] ADS\EXAM\CIRCULAR.C 2=[↕]
115:2
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

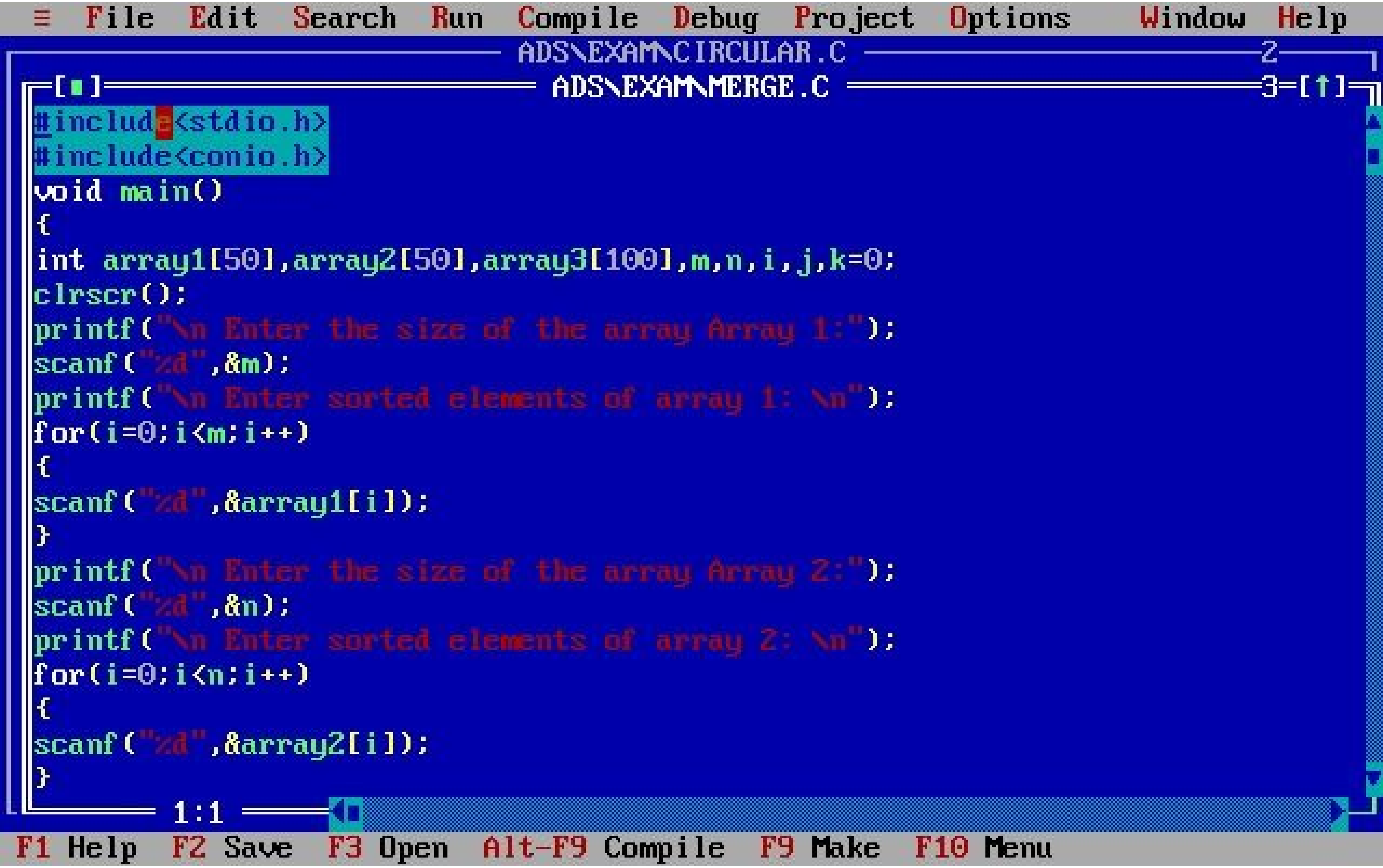
```
case 1:
printf("Insert the element for insertion in queue:");
scanf("%d",&item);
insert(item);
break;
case 2:
deletion();
break;
case 3:
display();
break;
case 4:
exit();
break;
default:
printf("Wrong choice\n");
}
}
while(choice!=4);
getch();
}
```



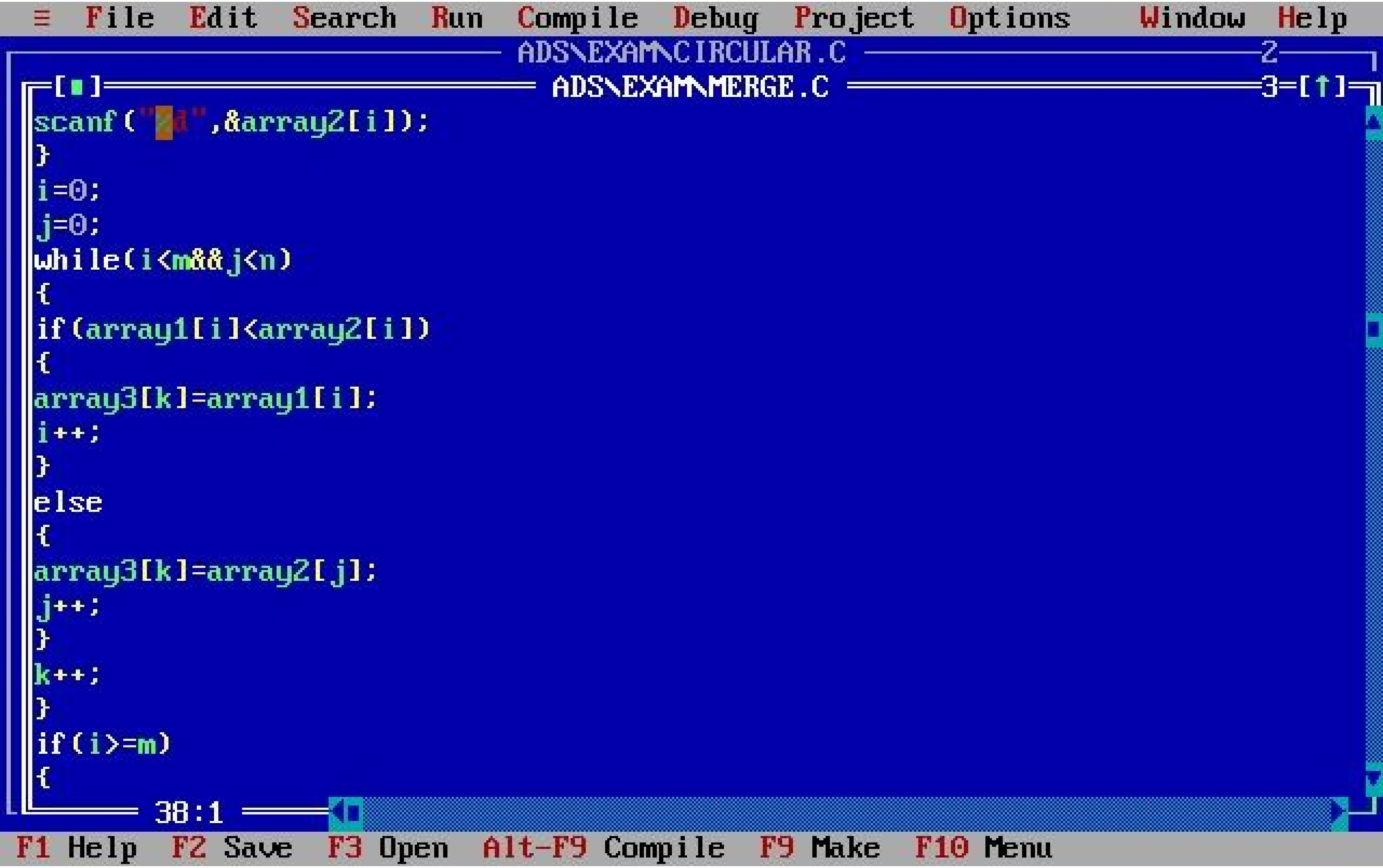
```
***Circular Queue***
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice:1
Insert the element for insertion in queue:12

***Circular Queue***
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice:3
Queue elements:12

***Circular Queue***
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice:
```



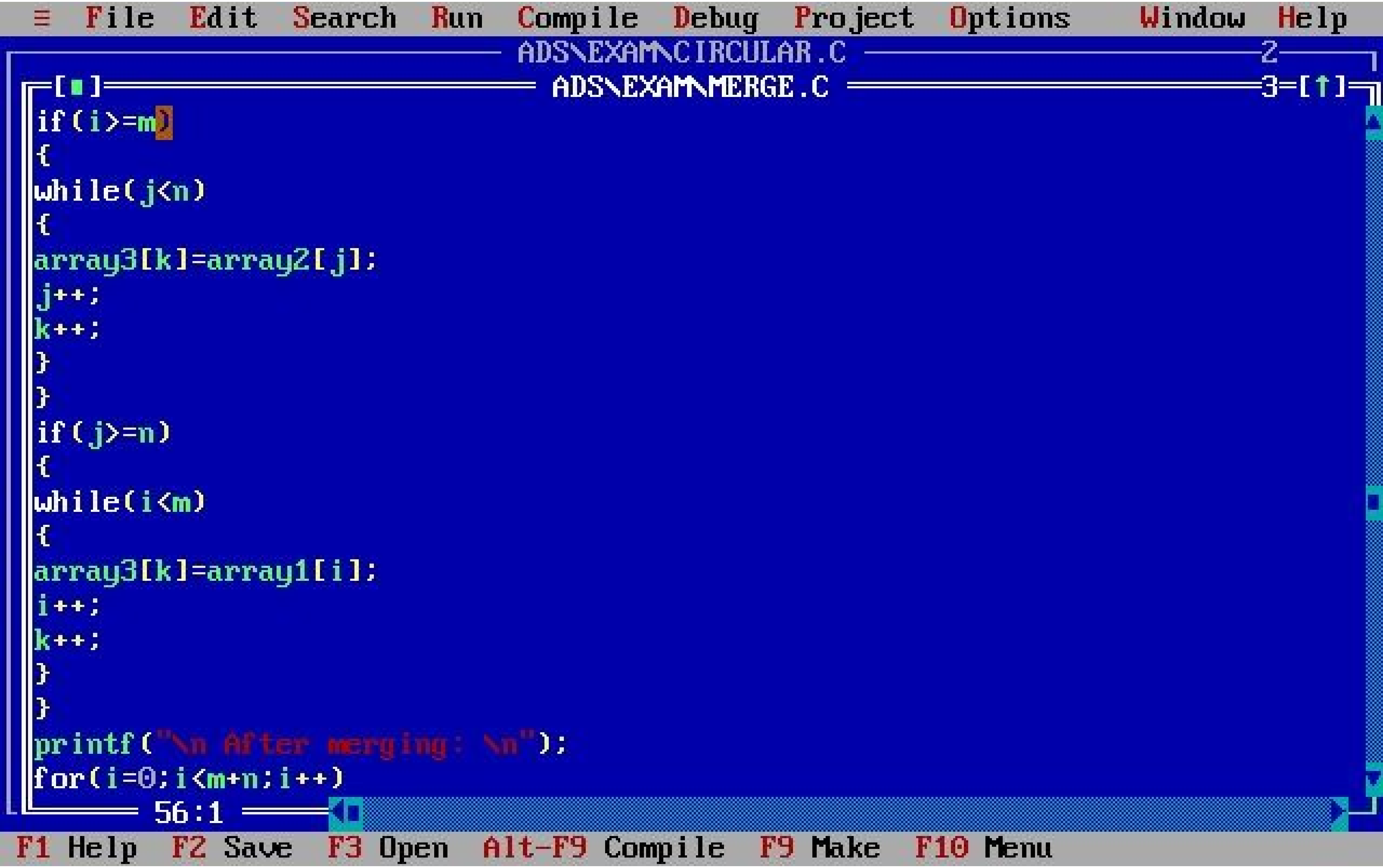
```
Enter your choice:1
Insert the element for insertion in queue:12

***Circular Queue***
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice:3
Queue elements:12

***Circular Queue***
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice:2
Element deleted from queue is:12

***Circular Queue***
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice:4
```


≡ File Edit Search Run Compile Debug Project Options Window Help
ADS\EXAM\CIRCULAR.C
2
[■]
ADS\EXAM\MERGE.C
3 [↑]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

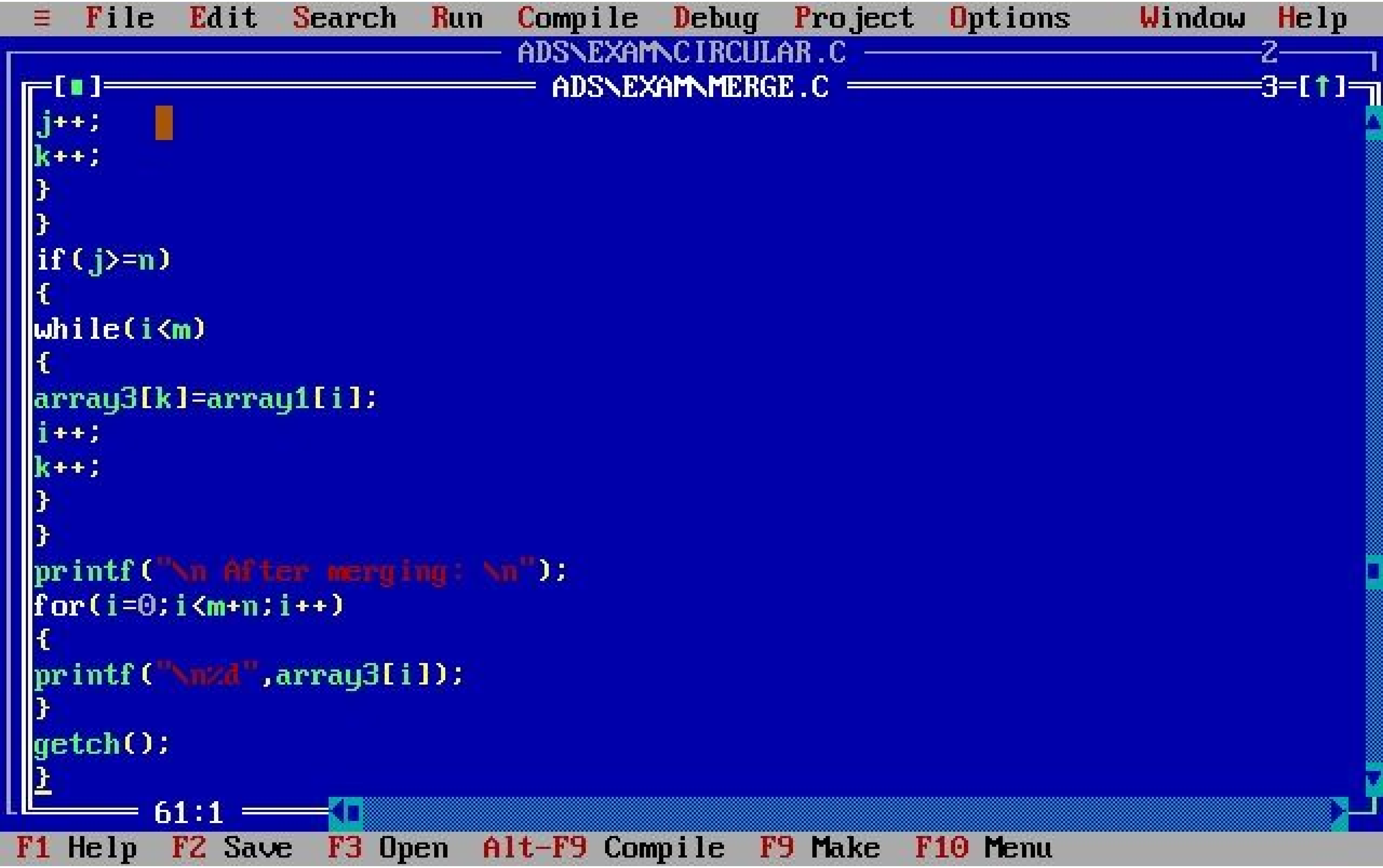
```
#include<stdio.h>
#include<conio.h>
void main()
{
int array1[50],array2[50],array3[100],m,n,i,j,k=0;
clrscr();
printf("\n Enter the size of the array Array 1:");
scanf("%d",&m);
printf("\n Enter sorted elements of array 1: \n");
for(i=0;i<m;i++)
{
scanf("%d",&array1[i]);
}
printf("\n Enter the size of the array Array 2:");
scanf("%d",&n);
printf("\n Enter sorted elements of array 2: \n");
for(i=0;i<n;i++)
{
scanf("%d",&array2[i]);
}
```


≡ File Edit Search Run Compile Debug Project Options Window Help
ADS\EXAM\CIRCULAR.C
2
[■]
ADS\EXAM\MERGE.C
3
[↑]
38:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

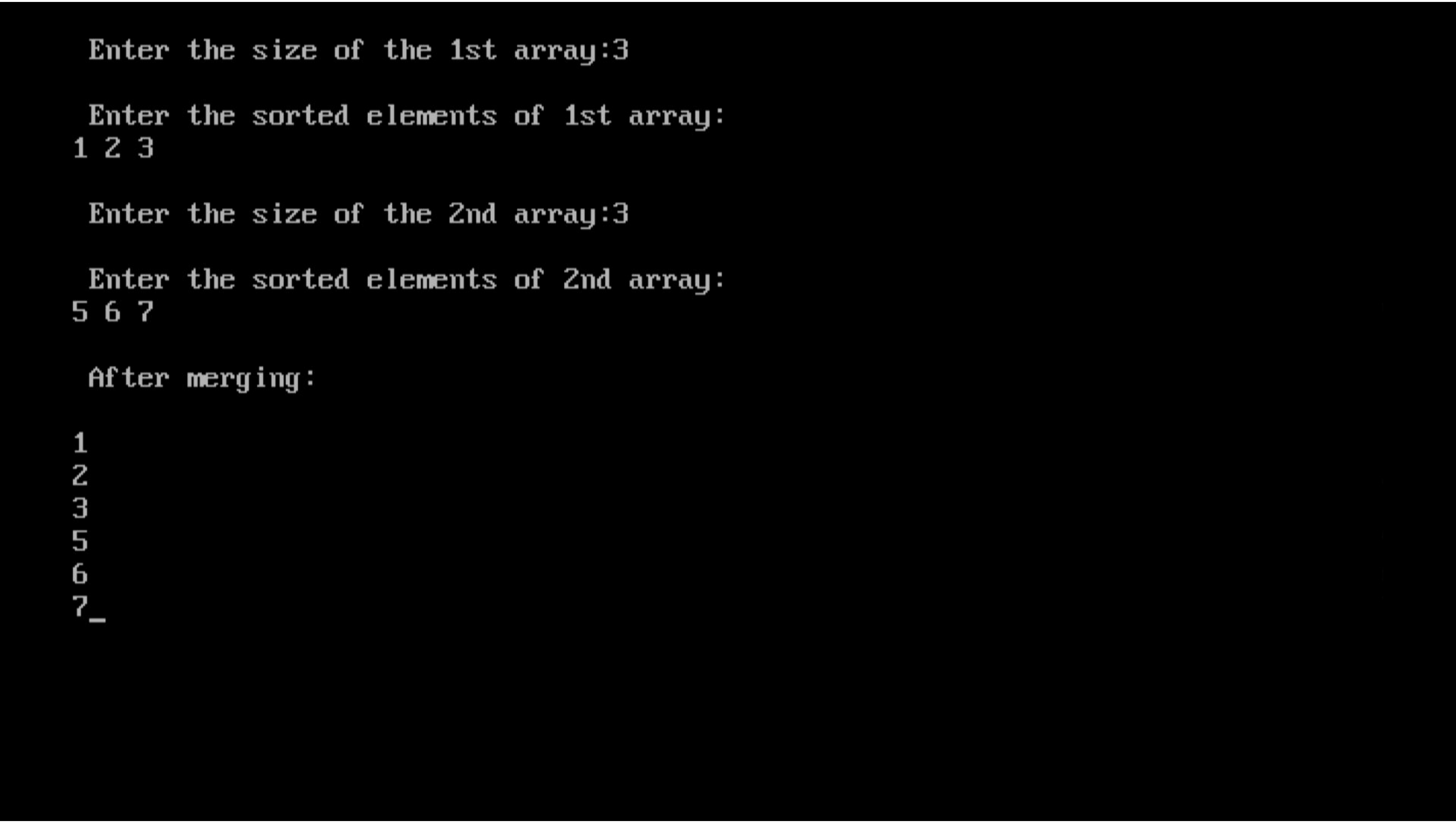
```
scanf("%d",&array2[i]);
}
i=0;
j=0;
while(i<m&&j<n)
{
if(array1[i]<array2[i])
{
array3[k]=array1[i];
i++;
}
else
{
array3[k]=array2[j];
j++;
}
k++;
}
if(i>=m)
{
```


≡ File Edit Search Run Compile Debug Project Options Window Help
ADS\EXAM\CIRCULAR.C
2
[■]
ADS\EXAM\MERGE.C
3=[↑]
56:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
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");
for(i=0;i<m+n;i++)
```


≡ File Edit Search Run Compile Debug Project Options Window Help
ADS\EXAM\CIRCULAR.C
2
[■]
ADS\EXAM\MERGE.C
3=[↑]
61:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

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



```
 Enter the size of the 1st array:3

 Enter the sorted elements of 1st array:
1 2 3

 Enter the size of the 2nd array:3

 Enter the sorted elements of 2nd array:
5 6 7

 After merging:

1
2
3
5
6
7
```