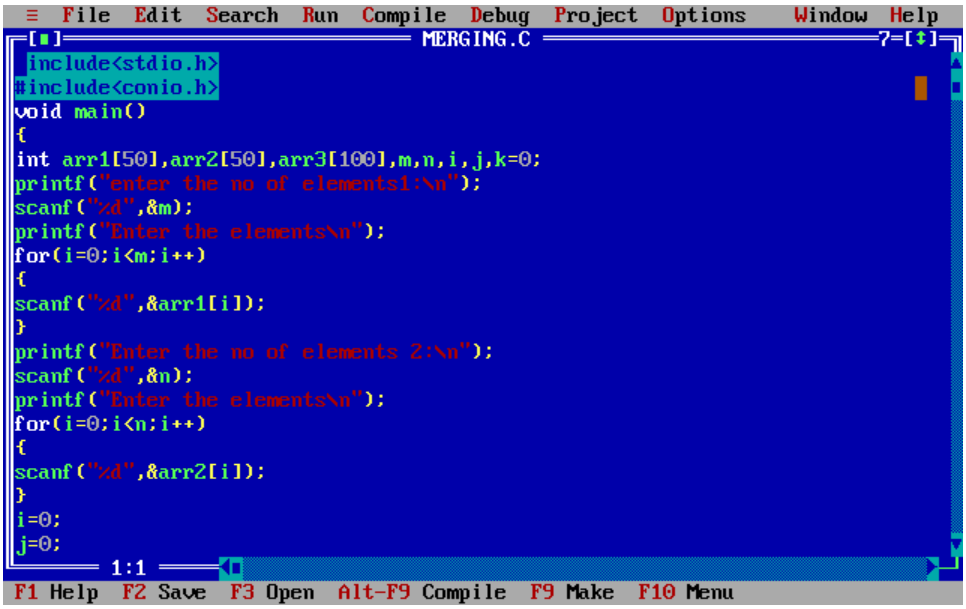
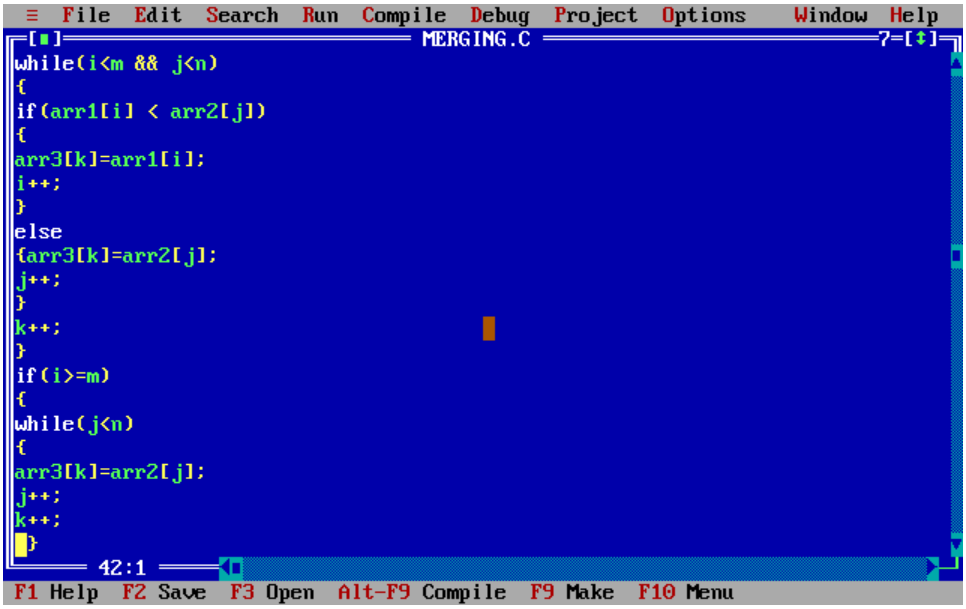
Program 1:

Merge two arrays

PROGRAM CODE:





```
 File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
                              MERGING.C                                 
while(j<n)
{
arr3[k]=arr2[j];
j++;
k++;
}}
if(j>=n)
{
while(i<m)
{
arr3[k]=arr1[i];
i++;
j++;
}
}printf("Merged array:\n");
for(i=0;i<m+n;i++)
{
printf("%d\n",arr3[i]);
}
getch();
                57:1
F1 Help  F2 Save  F3 Open  Alt-F9 Compile  F9 Make  F10 Menu
```

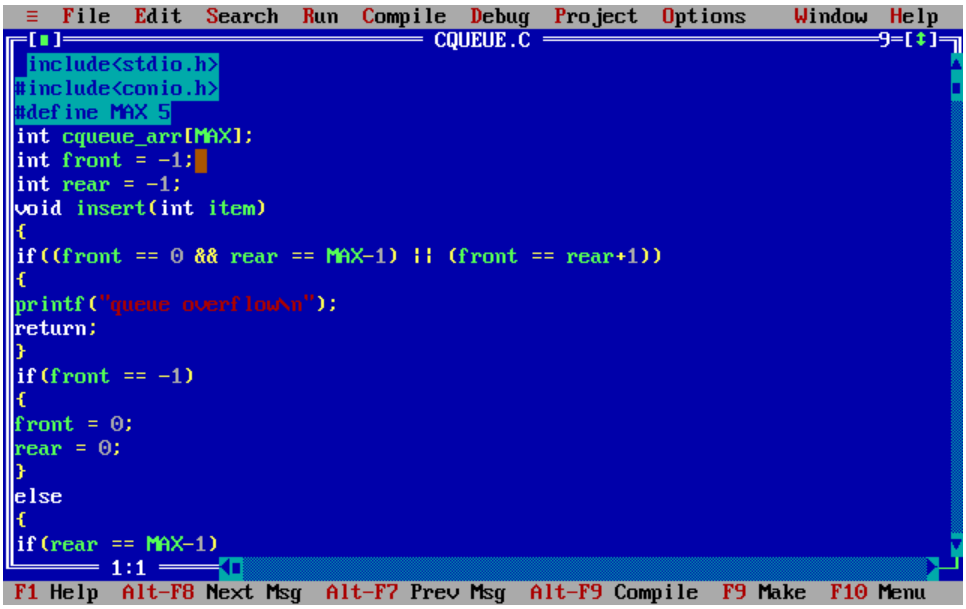
OUTPUT:

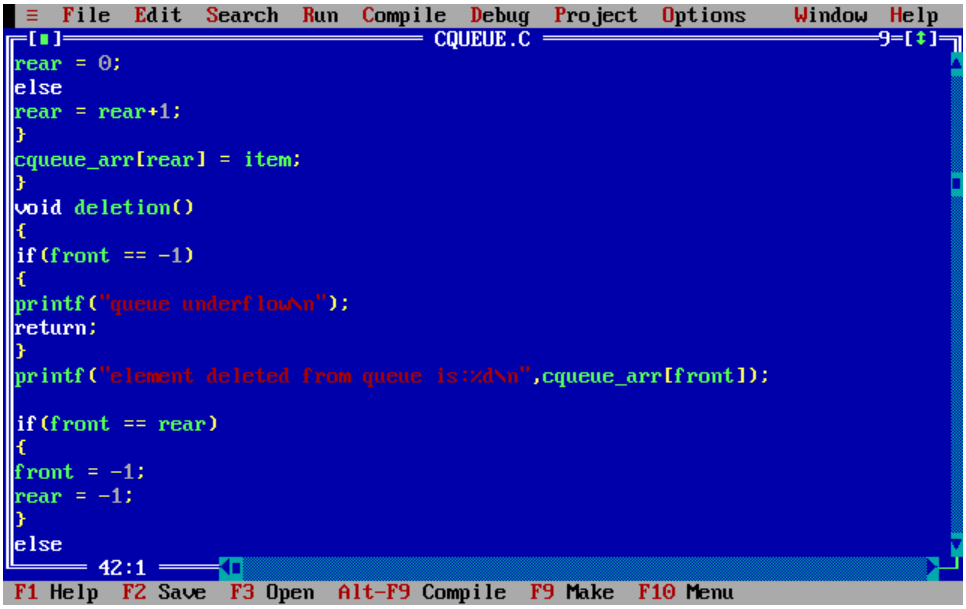
```
C:\TC\BIN>tc.exe
enter the no of elements1:
3
Enter the elements
1 2 3
Enter the no of elements 2:
4
Enter the elements
5 6 7 9
Merged array:
1
2
3
5
6
7
9
```

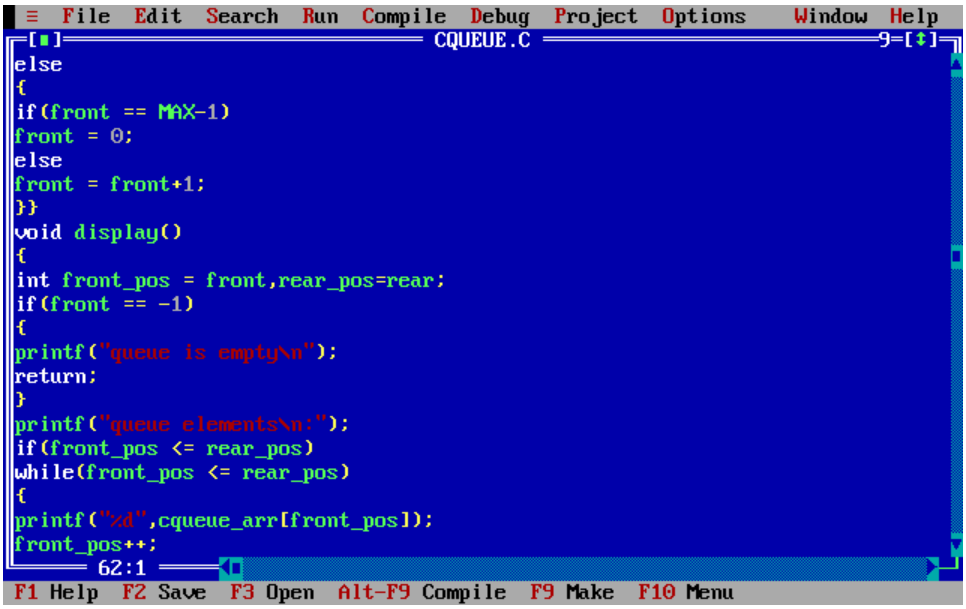
Program 2:

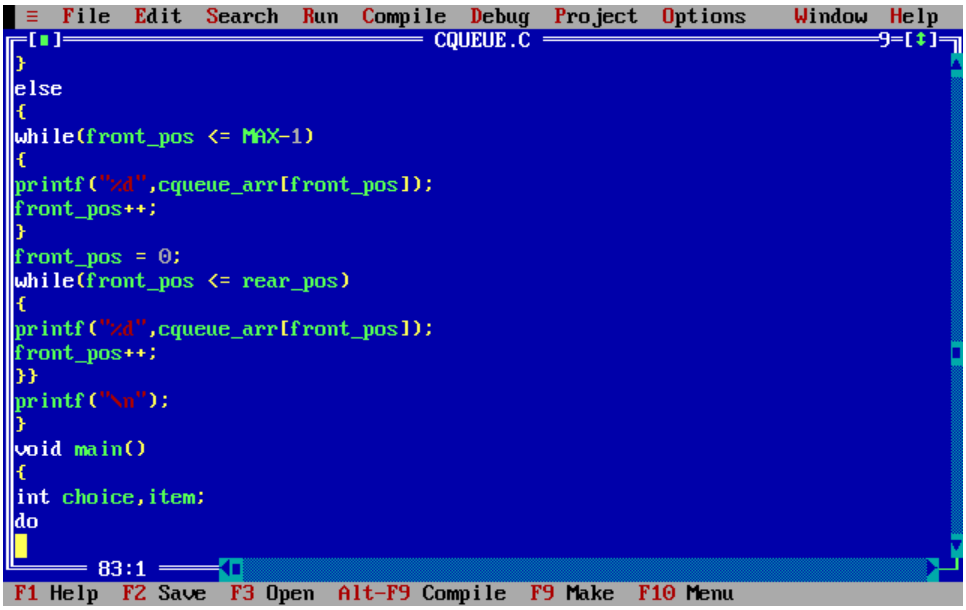
Implement circular queue

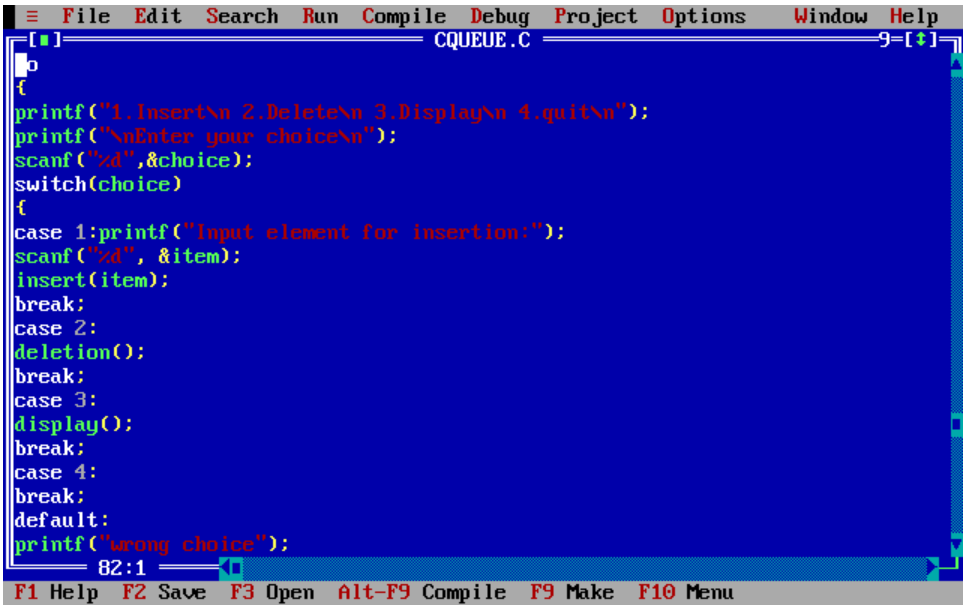
PROGRAM CODE:

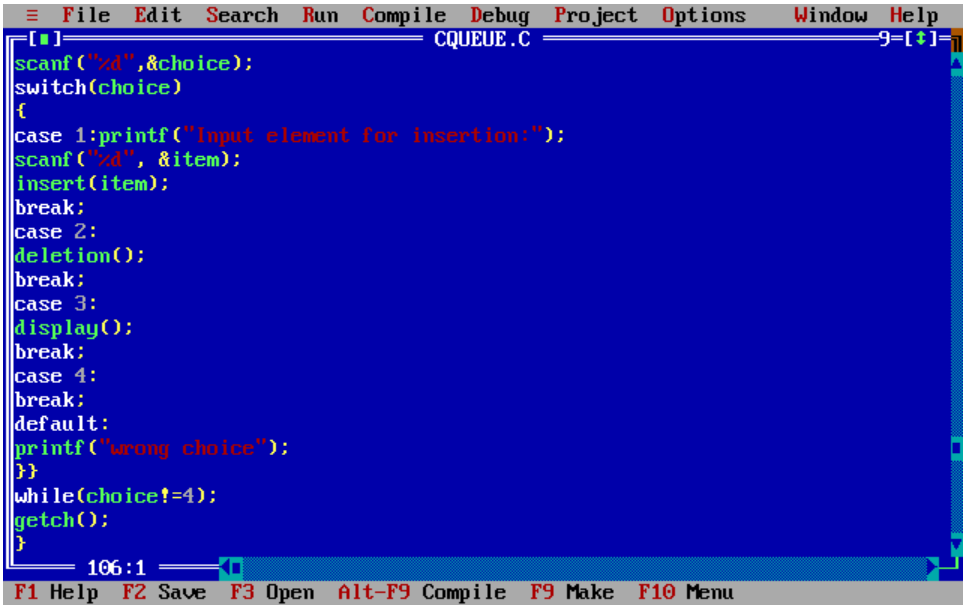

File Edit Search Run Compile Debug Project Options Window Help
CQUEUE.C
#include<stdio.h>
#include<conio.h>
#define MAX 5
int cqueue_arr[MAX];
int front = -1;
int rear = -1;
void insert(int item)
{
if((front == 0 && rear == MAX-1) || (front == rear+1))
{
printf("queue overflow\n");
return;
}
if(front == -1)
{
front = 0;
rear = 0;
}
else
{
if(rear == MAX-1)
1:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
CQUEUE.C
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
42:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
CQUEUE.C
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\n:");
if(front_pos <= rear_pos)
while(front_pos <= rear_pos)
{
printf("%d",cqueue_arr[front_pos]);
front_pos++;
62:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
CQUEUE.C
}
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()
{
int choice,item;
do
83:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
CQUEUE.C
do
{
printf("1.Insert\n 2.Delete\n 3.Display\n 4.quit\n");
printf("\nEnter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("Input element for insertion:");
scanf("%d", &item);
insert(item);
break;
case 2:
deletion();
break;
case 3:
display();
break;
case 4:
break;
default:
printf("wrong choice");
82:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
CQUEUE.C
scanf("%d",&choice);
switch(choice)
{
case 1:printf("Input element for insertion:");
scanf("%d", &item);
insert(item);
break;
case 2:
deletion();
break;
case 3:
display();
break;
case 4:
break;
default:
printf("wrong choice");
}}
while(choice!=4);
getch();
}
106:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

OUTPUT:

```
1.Insert
2.Delete
3.Display
4.quit

Enter your choice
1
Input element for insertion:23
1.Insert
2.Delete
3.Display
4.quit

Enter your choice
1
Input element for insertion:25
1.Insert
2.Delete
3.Display
4.quit

Enter your choice
1
Input element for insertion:28
```

```
:232528
1.Insert
2.Delete
3.Display
4.quit

Enter your choice
2
element deleted from queue is:23
1.Insert
2.Delete
3.Display
4.quit

Enter your choice
3
queue elements
:2528
1.Insert
2.Delete
3.Display
4.quit

Enter your choice
4_
```