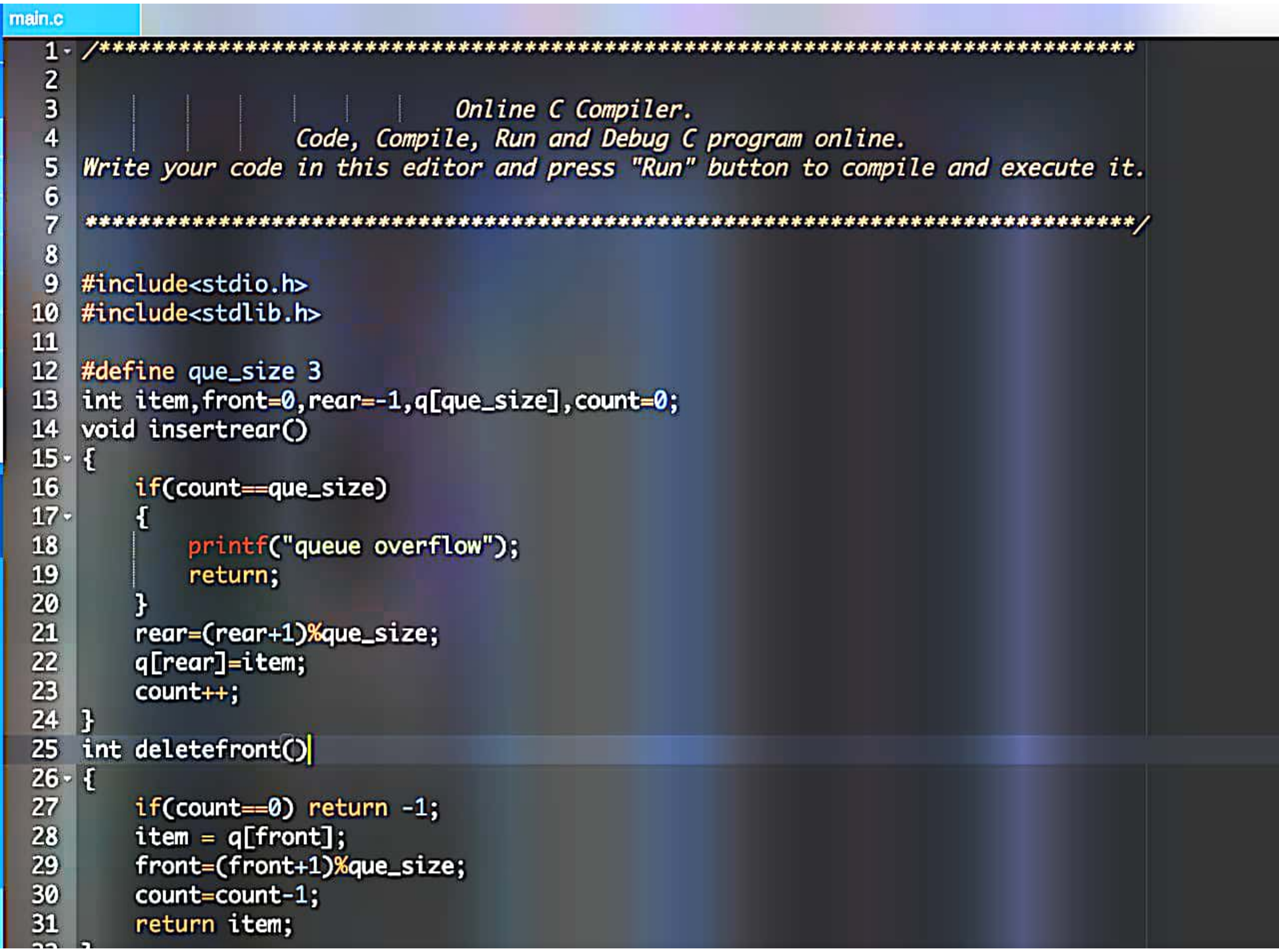
or →

Circular queue

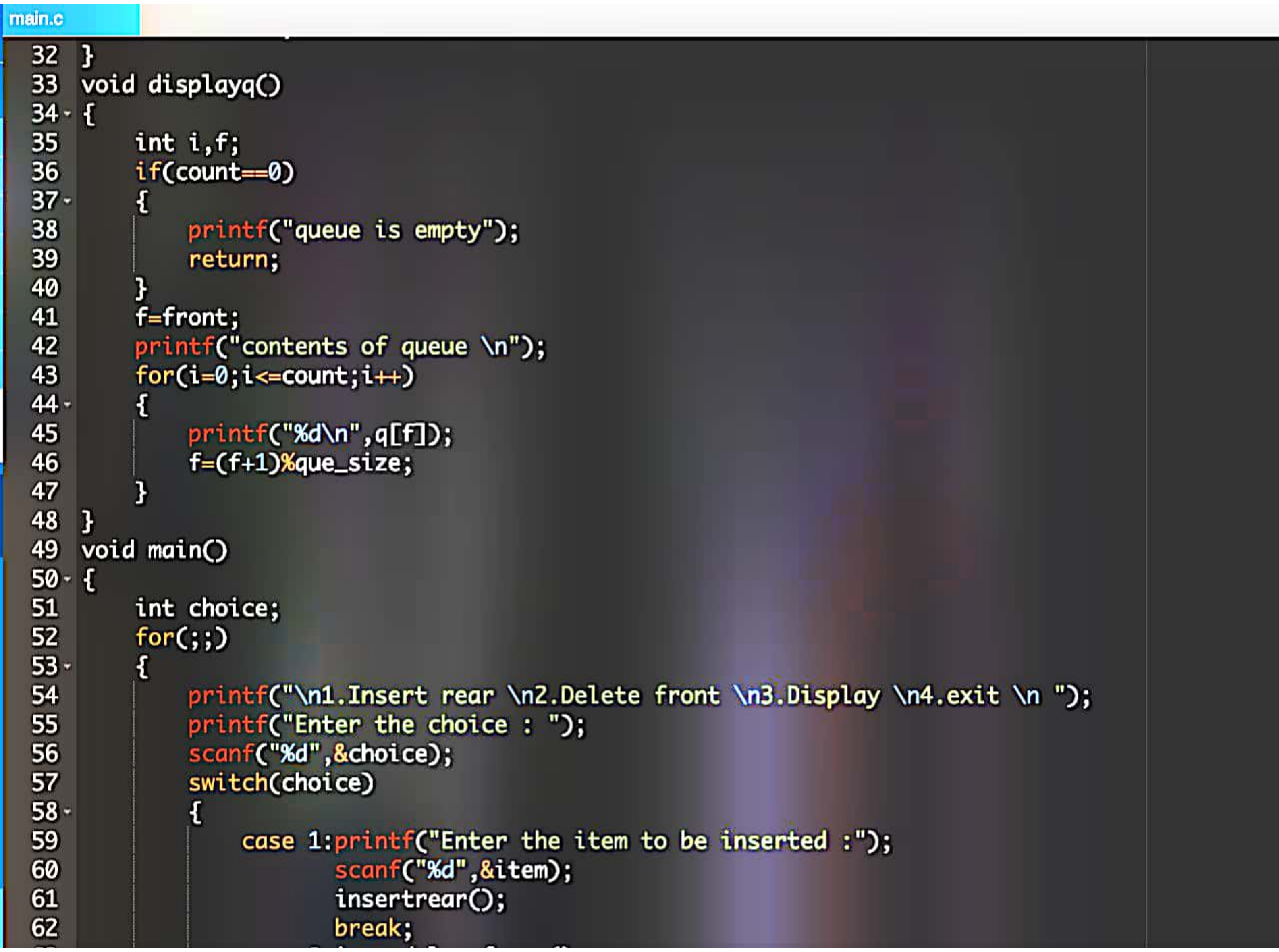
```
#include<stdio.h>
# include <stdlib.h>.
# include <Process.h>.
# define que-size 3.
int item, front=0, rear=-1, q[que-size], count=0;

Void insertRear(){.
   if (count==que-size)
   {
     Printf("queue overflows");
     return;
   }
   rear = (rear+1) % que-size;
   q[rear] = item;
   count++;
}
int deletefront() {.
   if(count==0) return -1;
   item = q[front];
   front = (front+1) % que-size;
   count = count-1;
   return item;
}.
```

```
void displayq(
    int i, f;
    if (count == 0) {
    pf("queue is empty");
     return;
  }
 f = front;
 pf("contents of queue \n");
   for(i=0; i<=count; i++){.
      pf("%d\n", q[f]))
       f = (f+1) % que-size;
  }
}
 void main(){.
   int choice;
  for(;;) {
pf("\n1. Insert rear \n2. Delete front \n3. Display
 \n 4. exit \n");
pf("Enter the choice : ");
sf("%d", &choice);
switch (choice).
{
  case 1: pf(" Enter the item to be inserted:");
         sf("%d", &item);
```

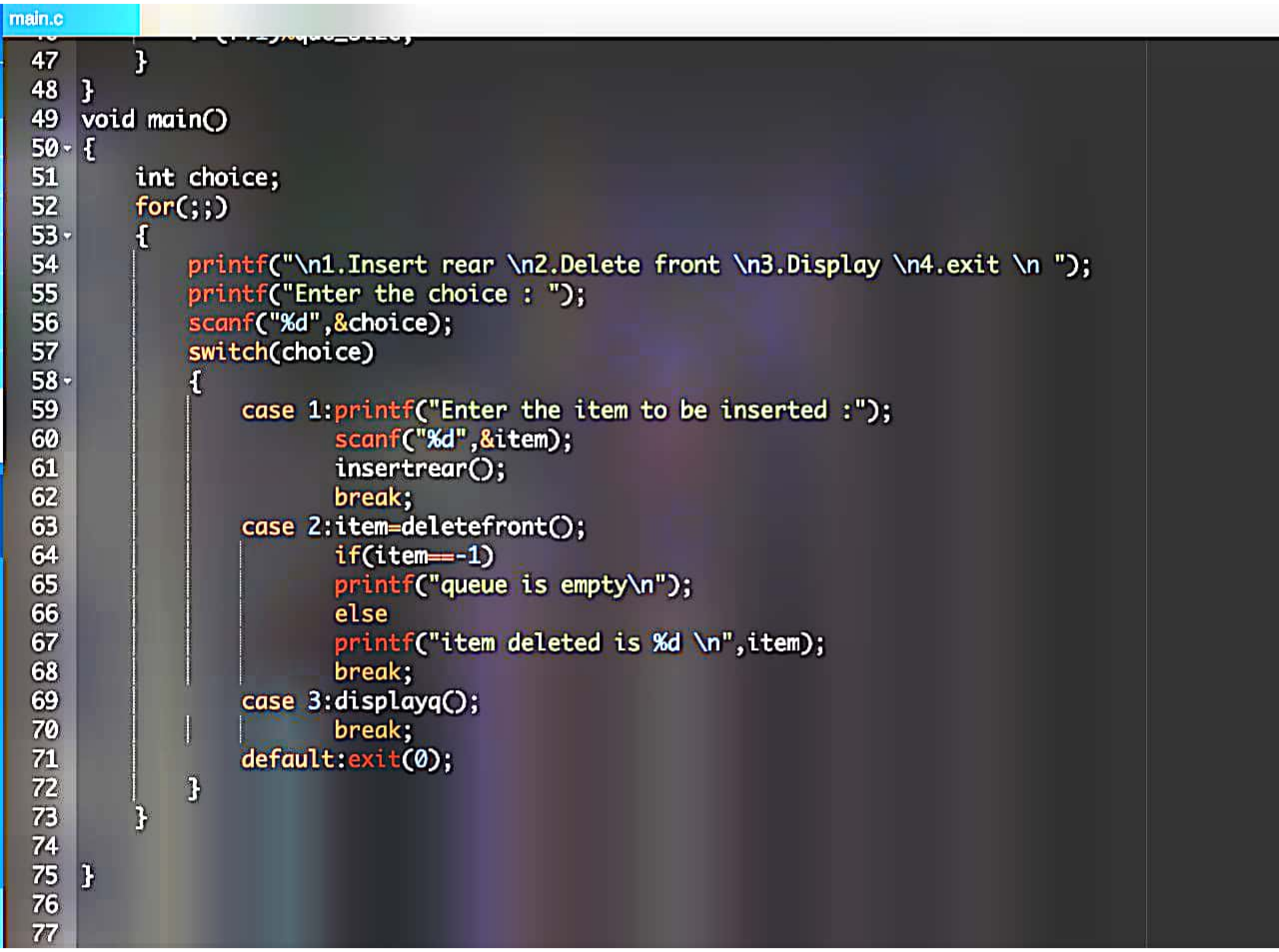
```
        insert rear();
         break;

case 2: item = delete front();
        if (item == -1)
        pf(" queue is empty");
       else
        pf (" item deleted is % d \n", item);
          break;

case 3: display q();
          break;

default : exit (0)0;

  }
 }
}.
```



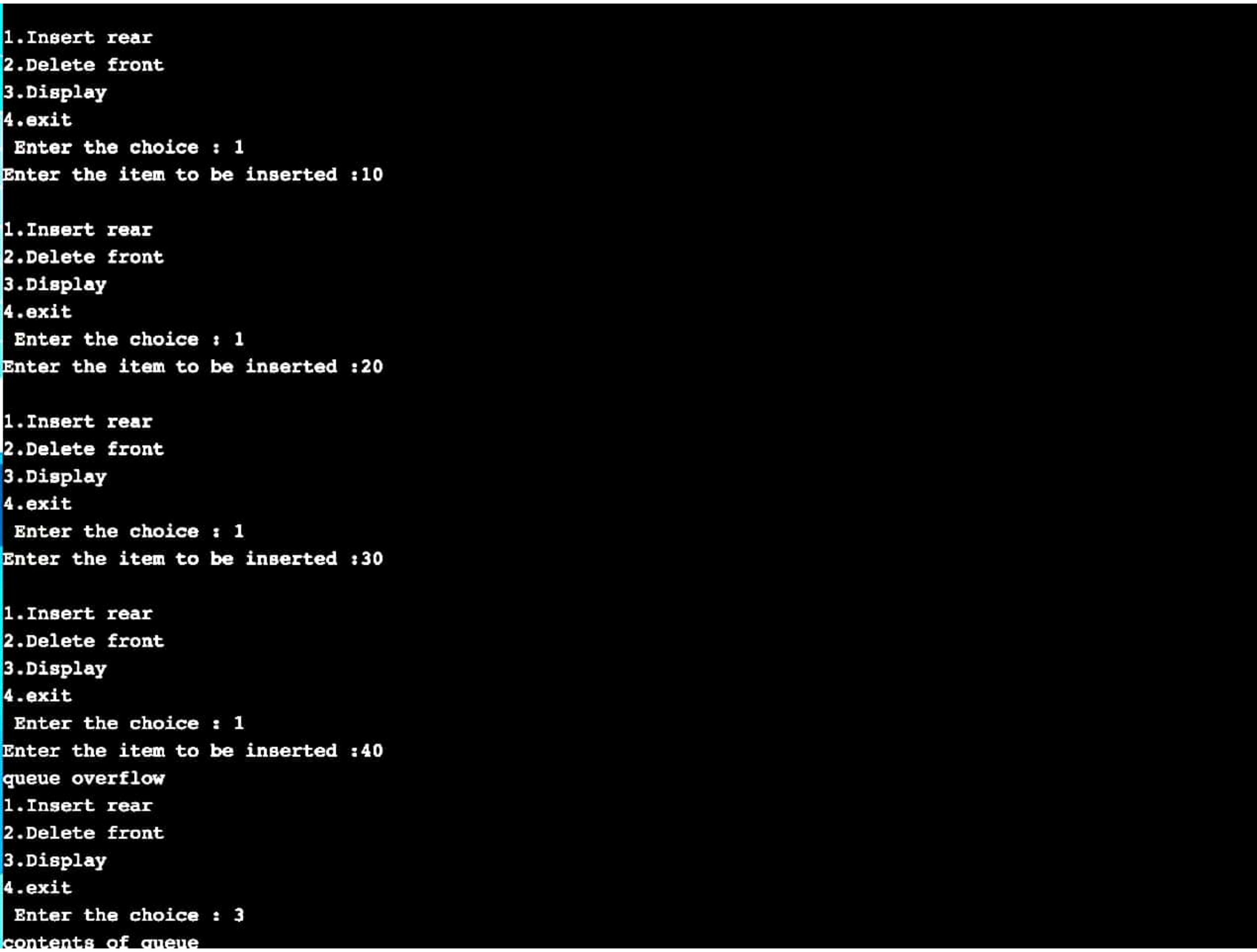
```c
/******************************************************************************

                            Online C Compiler.
                Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include<stdio.h>
#include<stdlib.h>

#define que_size 3
int item,front=0,rear=-1,q[que_size],count=0;
void insertrear()
{
    if(count==que_size)
    {
        printf("queue overflow");
        return;
    }
    rear=(rear+1)%que_size;
    q[rear]=item;
    count++;
}
int deletefront()
{
    if(count==0) return -1;
    item = q[front];
    front=(front+1)%que_size;
    count=count-1;
    return item;
```



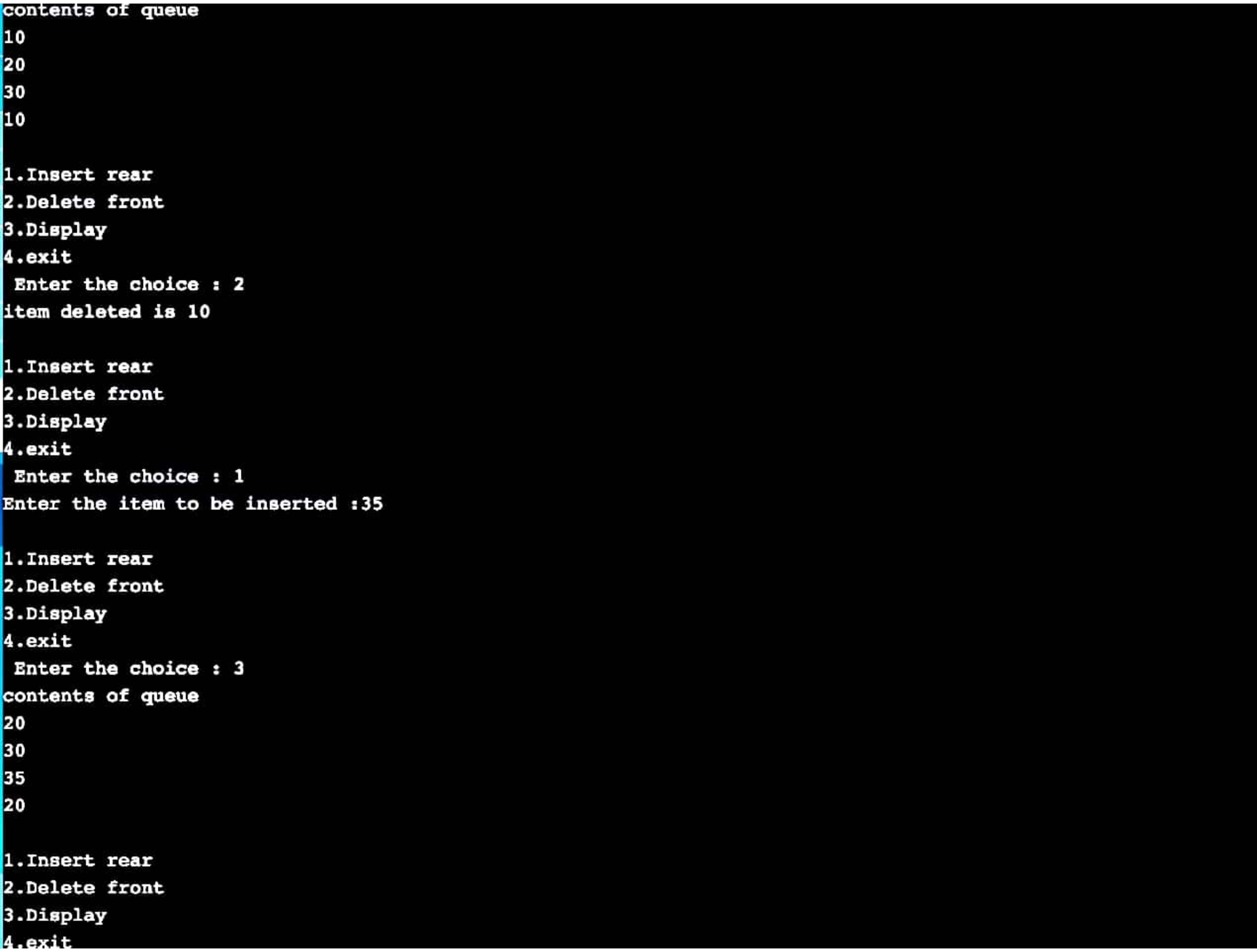
```c
}
void displayq()
{
    int i,f;
    if(count==0)
    {
        printf("queue is empty");
        return;
    }
    f=front;
    printf("contents of queue \n");
    for(i=0;i<=count;i++)
    {
        printf("%d\n",q[f]);
        f=(f+1)%que_size;
    }
}
void main()
{
    int choice;
    for(;;)
    {
        printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
        printf("Enter the choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter the item to be inserted :");
                   scanf("%d",&item);
                   insertrear();
                   break;
```



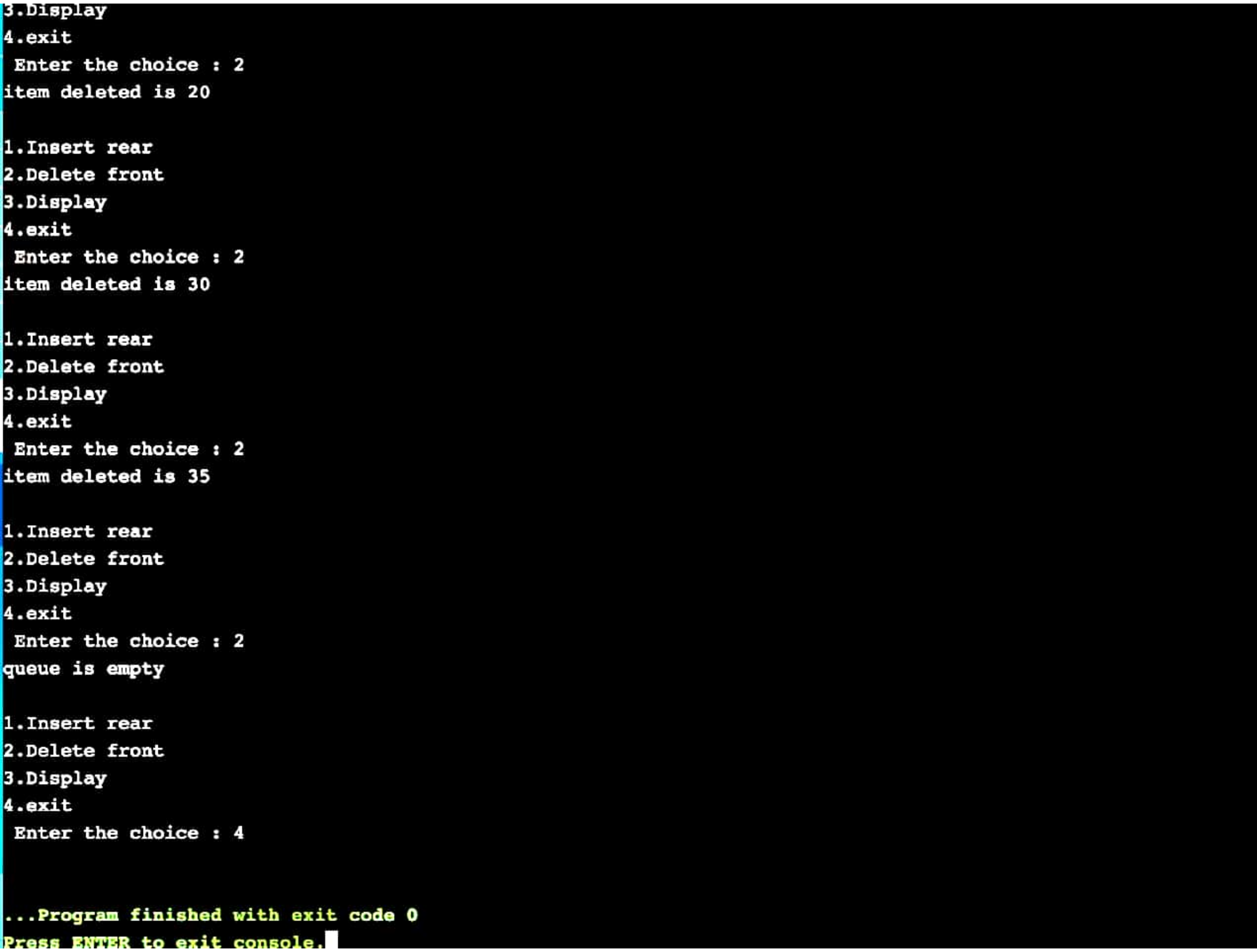
```c
    }
}
void main()
{
    int choice;
    for(;;)
    {
        printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
        printf("Enter the choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter the item to be inserted :");
                   scanf("%d",&item);
                   insertrear();
                   break;
            case 2:item=deletefront();
                   if(item==-1)
                   printf("queue is empty\n");
                   else
                   printf("item deleted is %d \n",item);
                   break;
            case 3:displayq();
                   break;
            default:exit(0);
        }
    }

}
```



```
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :10

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :20

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :30

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :40
queue overflow
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
```



```
contents of queue
10
20
30
10

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 10

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :35

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
20
30
35
20

1.Insert rear
2.Delete front
3.Display
4.exit
```



```
3.Display
4.exit
 Enter the choice : 2
item deleted is 20

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 30

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 35

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
queue is empty

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 4


...Program finished with exit code 0
Press ENTER to exit console.
```