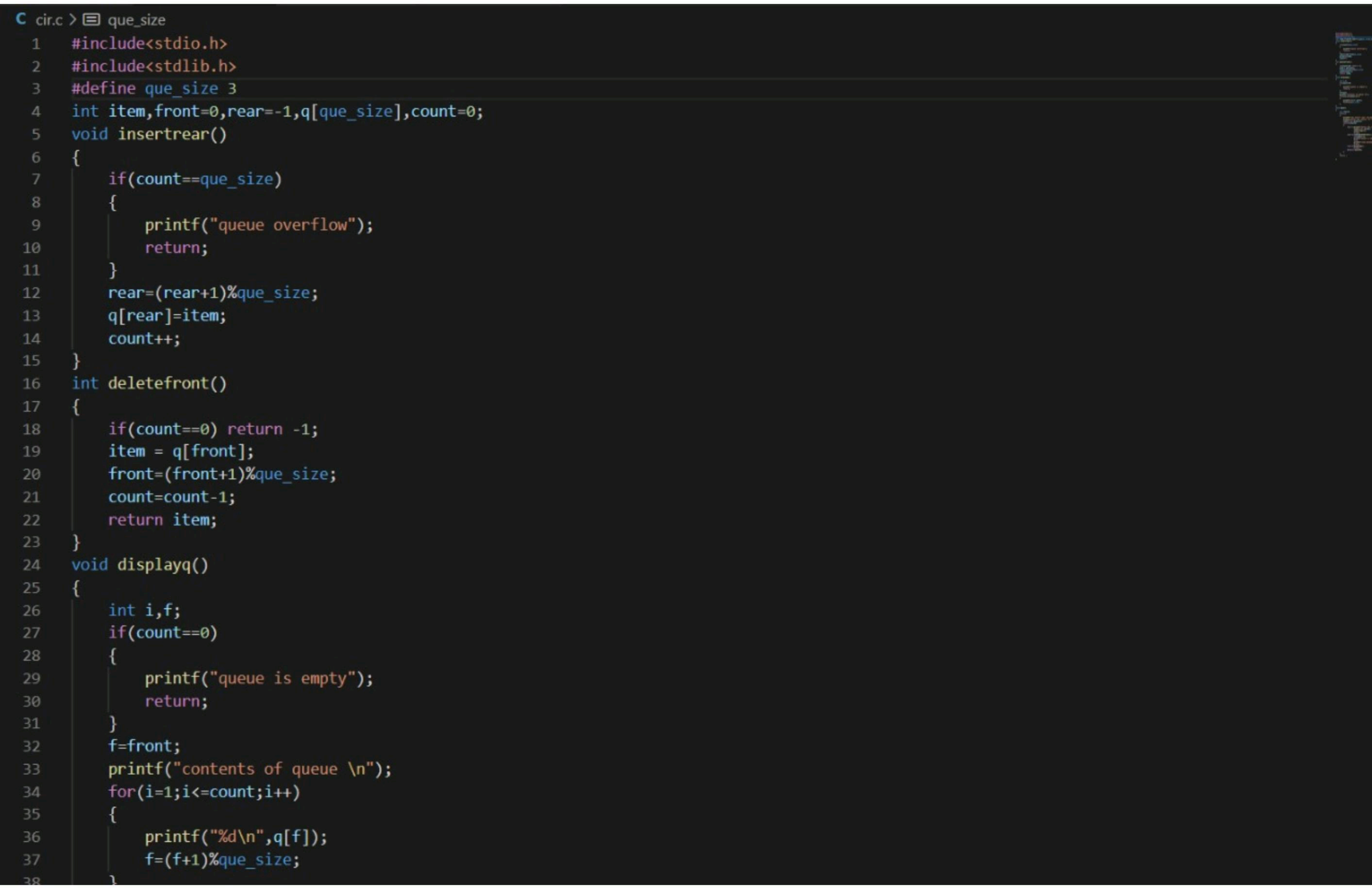
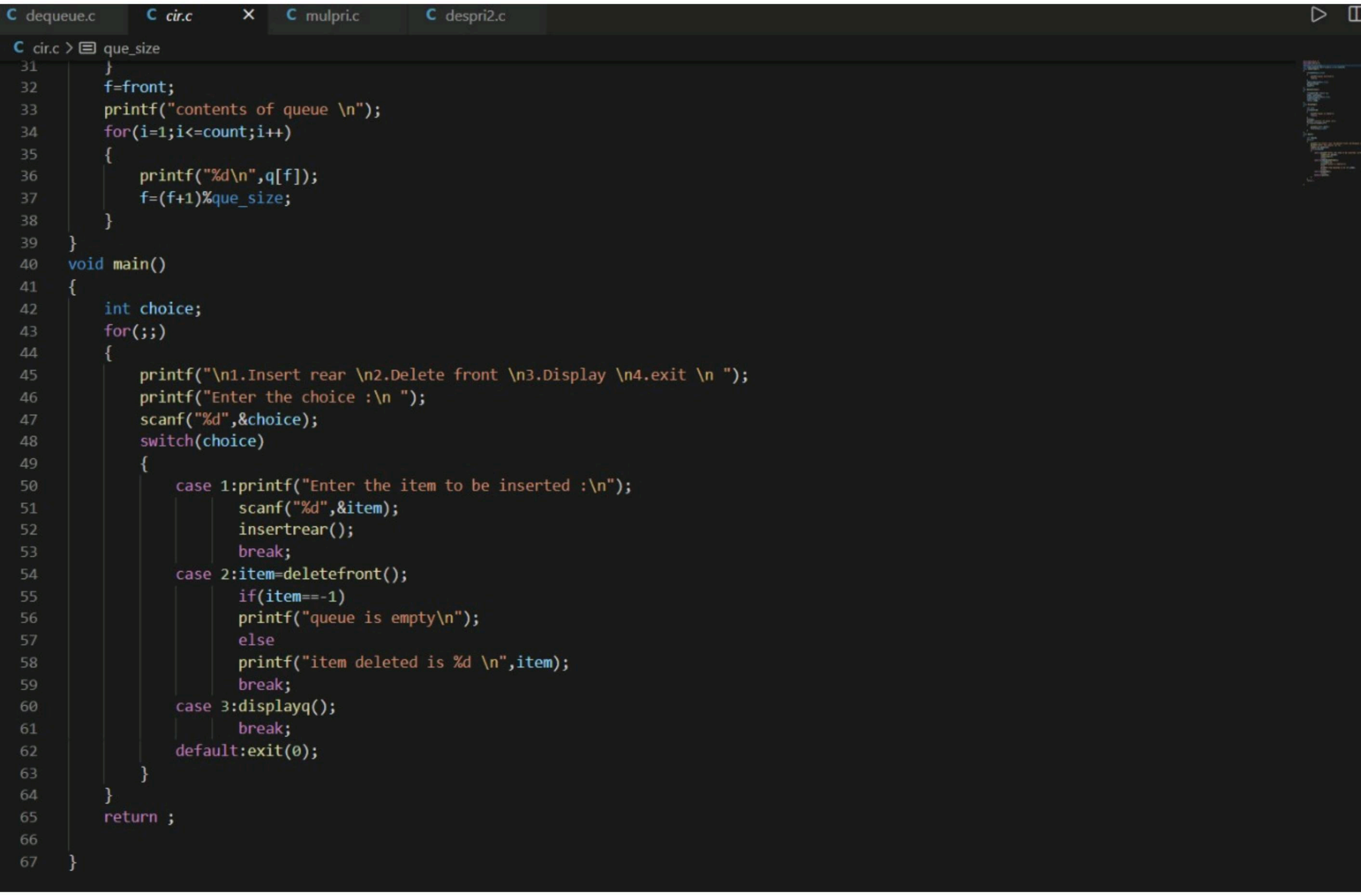
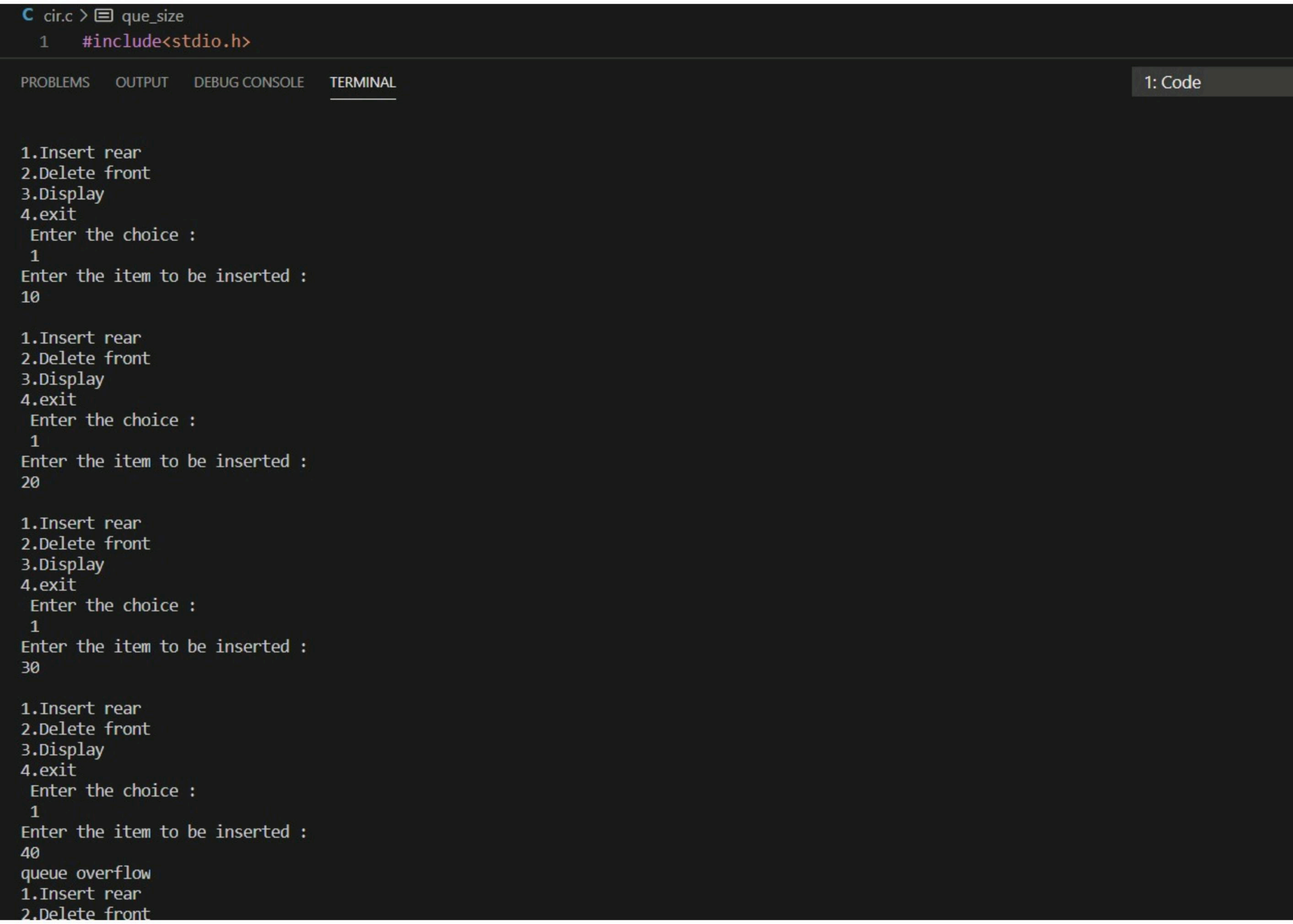


```c
#include<stdio.h>
#include<stdlib.h>
#define que_size 3
int item,front=0,rear=-1,q[que_size],count=0;
void insertrear()
{
    if(count==que_size)
    {
        printf("queue overflow");
        return;
    }
    rear=(rear+1)%que_size;
    q[rear]=item;
    count++;
}
int deletefront()
{
    if(count==0) return -1;
    item = q[front];
    front=(front+1)%que_size;
    count=count-1;
    return item;
}
void displayq()
{
    int i,f;
    if(count==0)
    {
        printf("queue is empty");
        return;
    }
    f=front;
    printf("contents of queue \n");
    for(i=1;i<=count;i++)
    {
        printf("%d\n",q[f]);
        f=(f+1)%que_size;
    }
```



```c
    }
    f=front;
    printf("contents of queue \n");
    for(i=1;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 :\n ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter the item to be inserted :\n");
                   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);
        }
    }
    return ;

}
```



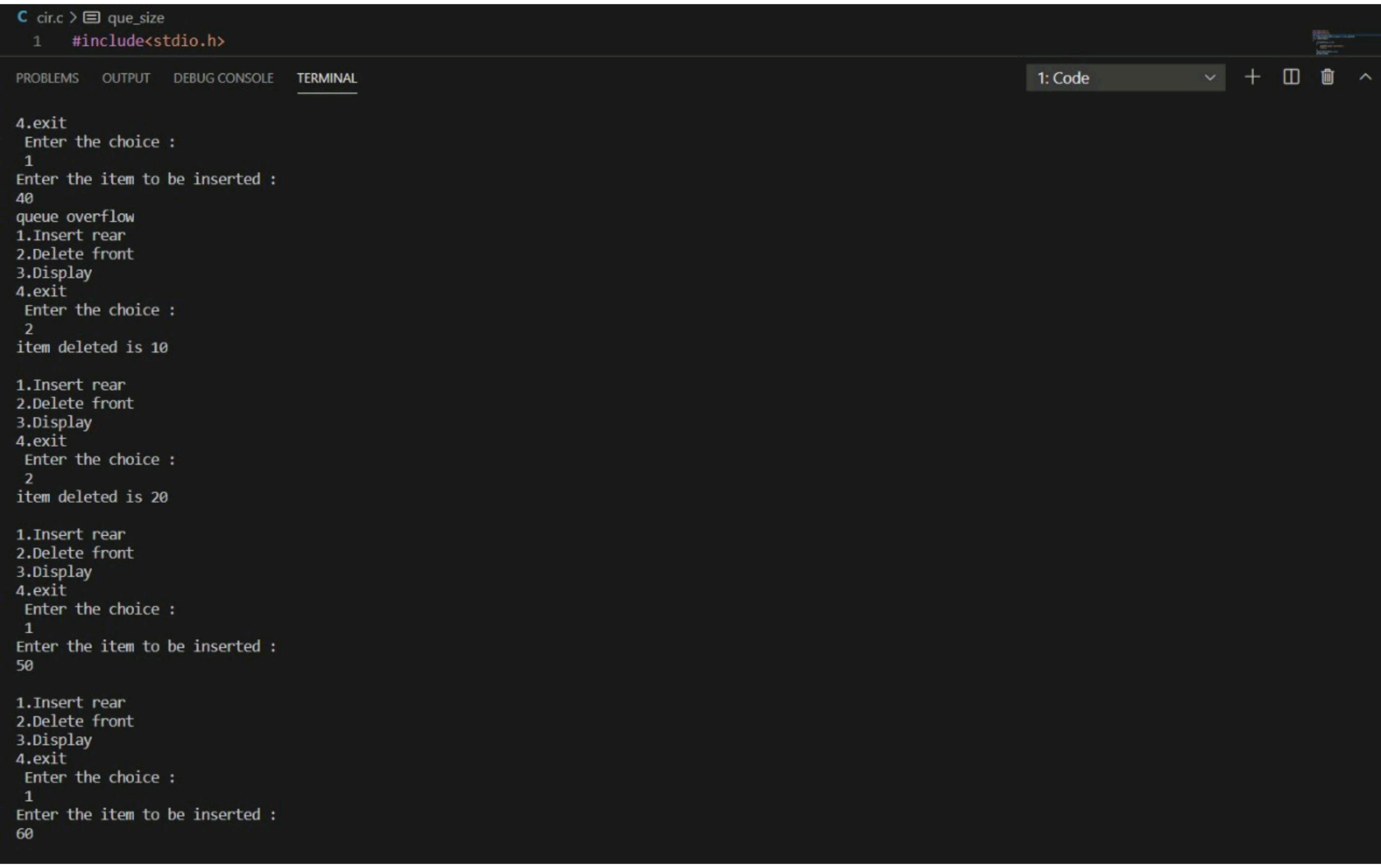
```
#include<stdio.h>
```

```
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
```



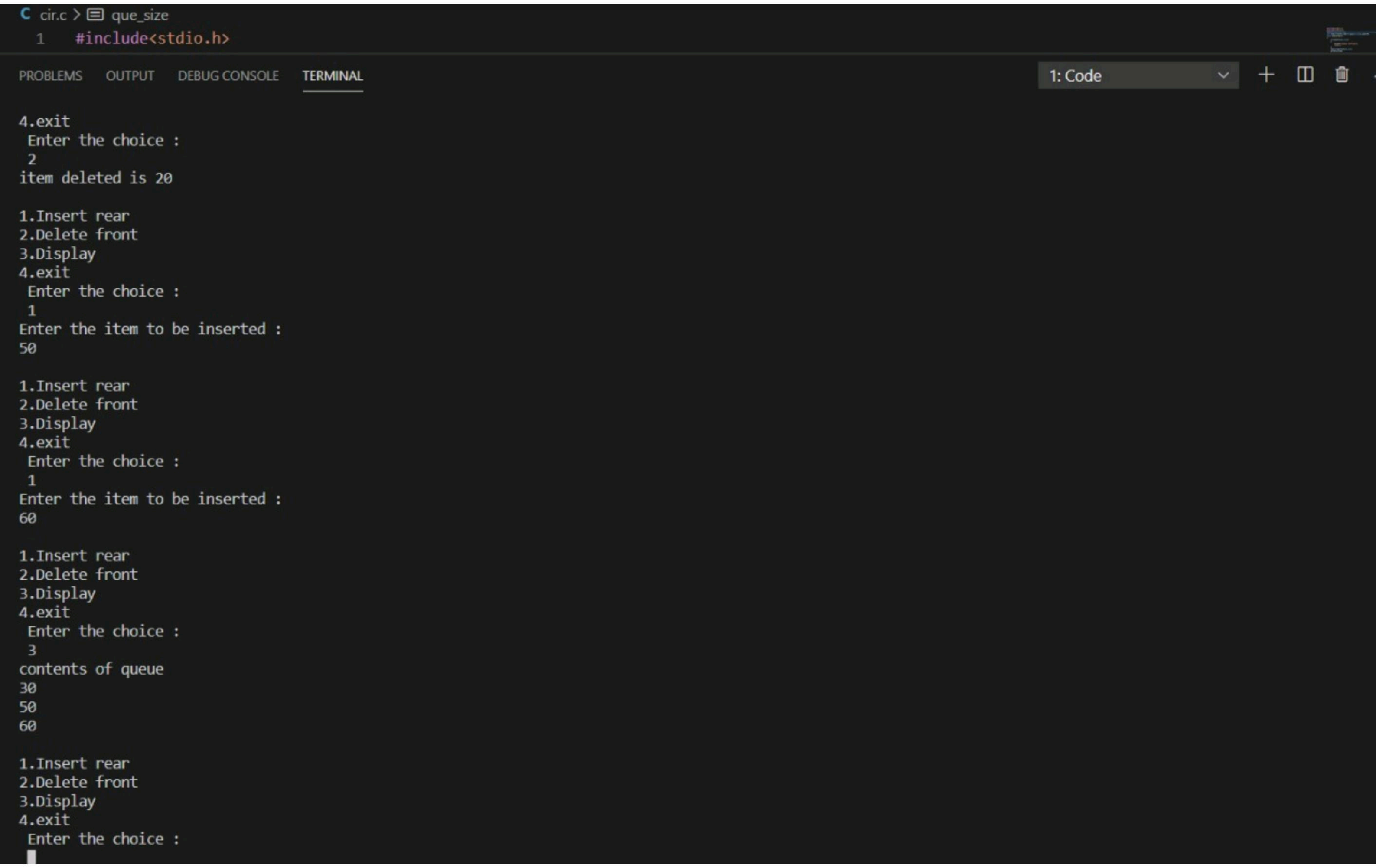
```
#include<stdio.h>
```

```
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 :
 2
item deleted is 10

1.Insert rear
2.Delete front
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 :
 1
Enter the item to be inserted :
50

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice :
 1
Enter the item to be inserted :
60
```



```
#include<stdio.h>
```

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

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice :
 1
Enter the item to be inserted :
50

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice :
 1
Enter the item to be inserted :
60

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice :
 3
contents of queue
30
50
60

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

CIRCULAR QUEUE

```
# include<stdio.h>
# include <conio.h>
# define Qs 5
int item , count=0, front =0, rear =-1, q [QS];
 void insertrear ( )
 {
 if (count ==Qs)
 {
   printf ("queue overflow");
    return;
  }
  rear = (rear + 1) % Qs ;
  q[rear] = item ;
   count++ ;
  }
  int delete front( )
  {
   if (count = =0) return -1 ;
     item = q [front];
     front = (front +1) % Qs ;
      count = count-1 ;
      return item ;

   }
   void displayQ( )
   { int i,f ;
    if (count ==0)
    {
      printf ("the queue is empty");
      return ;
     }
     f = front ;
   printf (" the contents of the queues are:");
    for (i=front ; i<rear ; i++)
{
 printf ("%d\n", q[i]);
  f = (f+1) % Qs,
}
}
void main()
{ int choice;
 for (;;)
 { printf("\n1: insert from rear \n2: delete from front \n 3: display
        \n4: exct \n");
   printf ("Enter the choice \n");
    scanf("%d , choice);
    switch(choice)
   {
   case 1: printf("Enter the item to be inserted \n");
       scanf ("%d", item);
        insertrear();
        break;
       case 2: item = deletefront();
        if (item == -1)
        printf("queue is empty \n");
        else
        printf ("item deleted =%d ", item);
        break;
        case 3: display();
        break;
        default: exit(0);
     }
    }
   }
```