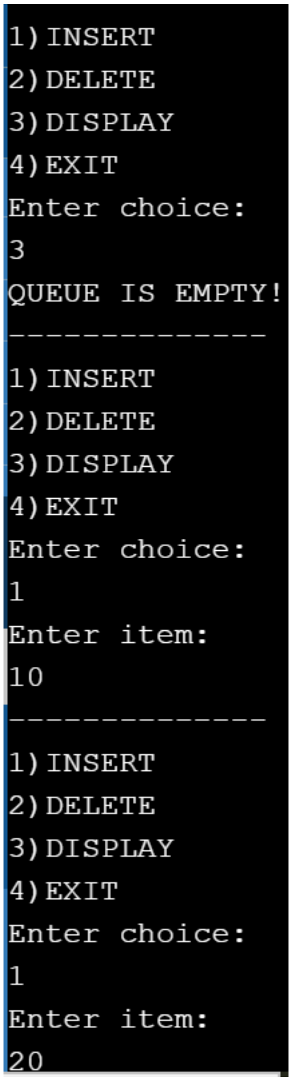
WEEK 6:(Circular queue)

CODE:

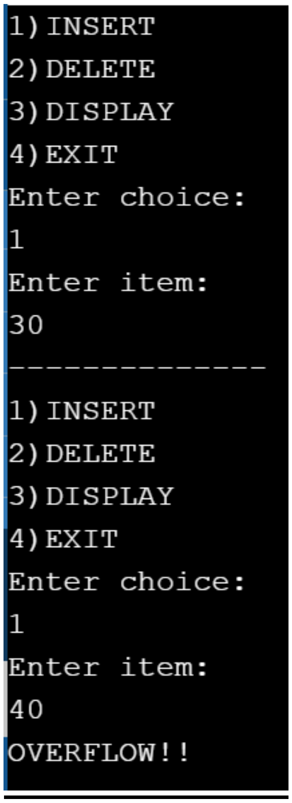
```
#include<stdio.h>

#define size 3

int item,f=0,r=-1,q[size],count=0;

void insertrear()

{

        if(count==size)

        {

                printf("OVERFLOW!!\n");

                return;

        }

        r=(r+1)%size;

        q[r]=item;

        count++;

}

int deletefront()

{

        if(count==0)return -1;

        item=q[f];

        f=(f+1)%size;

        count=count-1;

        return item;

}

void display()
```

```
{
    int i,f;
    if(count==0)
    {
        printf("QUEUE IS EMPTY!\n");return;
    }
    int front=f;
    printf("Contents:\n");
    for(int i=1;i<=count;i++)
    {
        printf("%d \n",q[front]);
        front=(front+1)%size;
    }
}
void main()
{
    int choice,check=1;
    while(check==1)
    {
        printf("--------------\n1)INSERT\n2)DELETE\n3)DISPLAY\n4)EXIT\nEnter choice:\n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter item:\n");scanf("%d",&item);insertrear();break;
            case 2:item=deletefront();
                    if(item==-1)printf("QUEUE IS EMPTY!\n");
                    else
                    printf("Deleted item:%d\n",item);break;
            case 3:display();break;
            default:check=0;


        }
    }
}
```

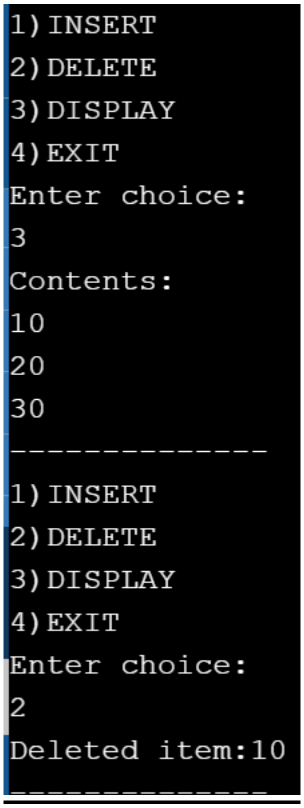
OUTPUT:



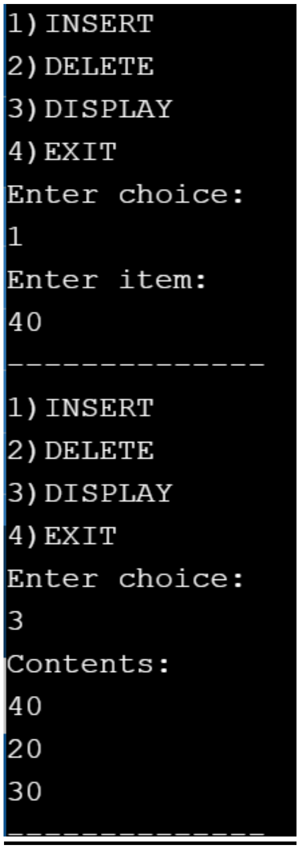
```
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
3
QUEUE IS EMPTY!
--------------
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
1
Enter item:
10
--------------
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
1
Enter item:
20
```



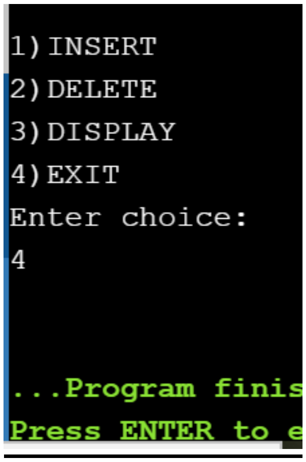
```
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
1
Enter item:
30
--------------
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
1
Enter item:
40
OVERFLOW!!
```



```
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
3
Contents:
10
20
30
--------------
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
2
Deleted item:10
--------------
```



```
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
1
Enter item:
40
--------------
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
3
Contents:
40
20
30
--------------
```



```
1)INSERT
2)DELETE
3)DISPLAY
4)EXIT
Enter choice:
4


...Program finis
Press ENTER to e
```

DOUBLE ENDED QUEUE:

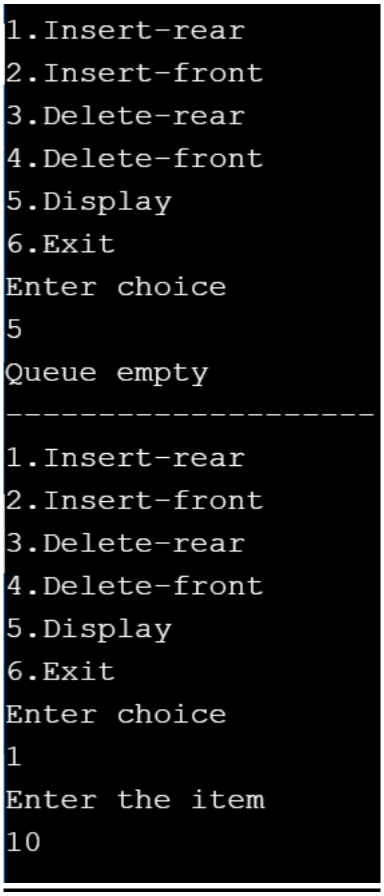
CODE:

```
#include <stdio.h>

#define size 3

int f=0,r=-1,ch;

int item,q[5];
int isfull()
{
    return(r==size-1)?1:0;
}
int isempty()
{
    return(f>r)?1:0;
}
void insert_rear()
{
    if(isfull())
    {
           printf("Queue overflow\n");
           return;
    }
    r=r+1;
    q[r]=item;
}
void delete_front()
{
    if(isempty())
          {
           printf("Queue empty\n");
           return;
          }
    printf("Item deleted is %d\n",q[(f)++]);
    if(f>r)
```

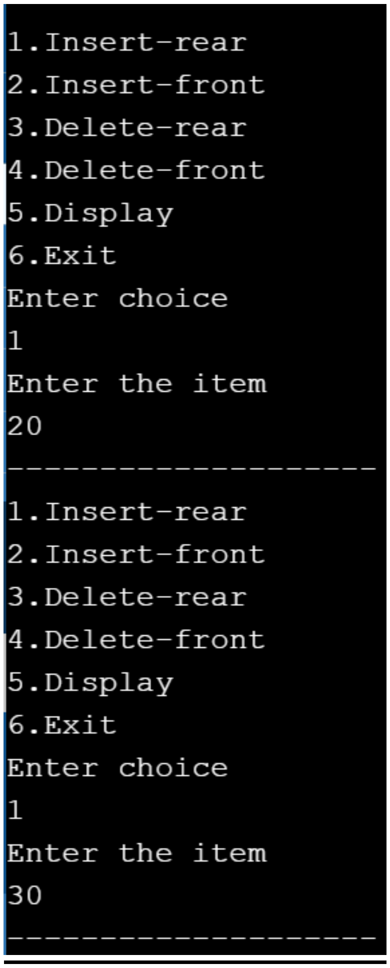
```
         {
           f=0;
           r=-1;
         }
}
void insert_front()
{
    if(f!=0)
    {
           f=f-1;
           q[f]=item;
           return;
    }
    else if((f==0)&&(r==-1))
    {
           q[++(r)]=item;
           return;
    }
    else
           printf("Insertion not possible\n");
}
void delete_rear()
{
    if(isempty())
         {
           printf("Queue is empty\n");
           f=0;
           r=-1;
           return;
          }
    printf("Item deleted is %d\n",q[(r)--]);


}
void display()
{
    int i;
    if(isempty())
          {
           printf("Queue empty\n");
           return;
          }
    for(i=f;i<=r;i++)
        printf("%d\n",q[i]);
}
void main()
{



   for(;;)
    {
        printf("--------------------\n1.Insert-rear\n2.Insert-front\n3.Delete-rear\n4.Delete-front\n5.Display\
n6.Exit\n");
        printf("Enter choice\n");
        scanf("%d",&ch);
        switch(ch)
           {
            case 1:printf("Enter the item\n");
```

```
                           scanf("%d",&item);

                           insert_rear();

                           break;

            case 2:printf("Enter the item\n");

                           scanf("%d",&item);

                           insert_front();

                           break;

            case 3:delete_rear();

                           break;

            case 4:delete_front();

                           break;

            case 5:display();

                           break;

            default:exit(0);

           }

        }


}
```

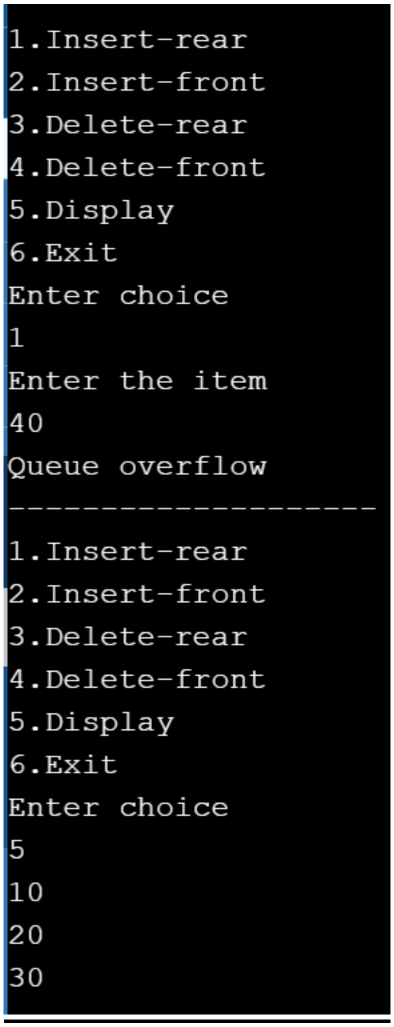
OUTPUT:



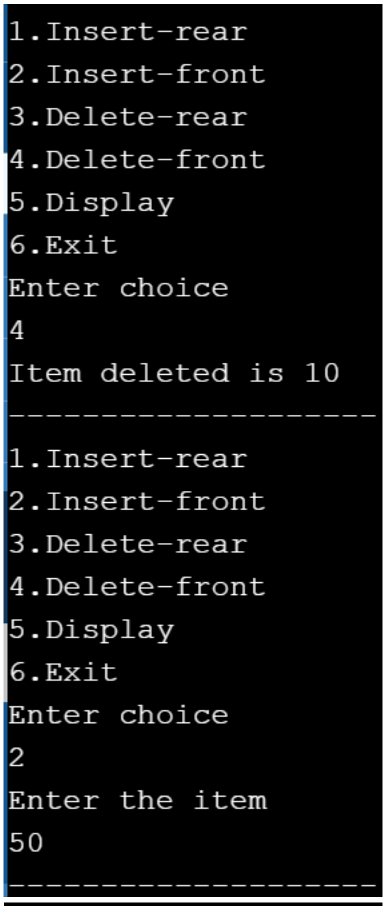
```
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
5
Queue empty
--------------------
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
1
Enter the item
10
```



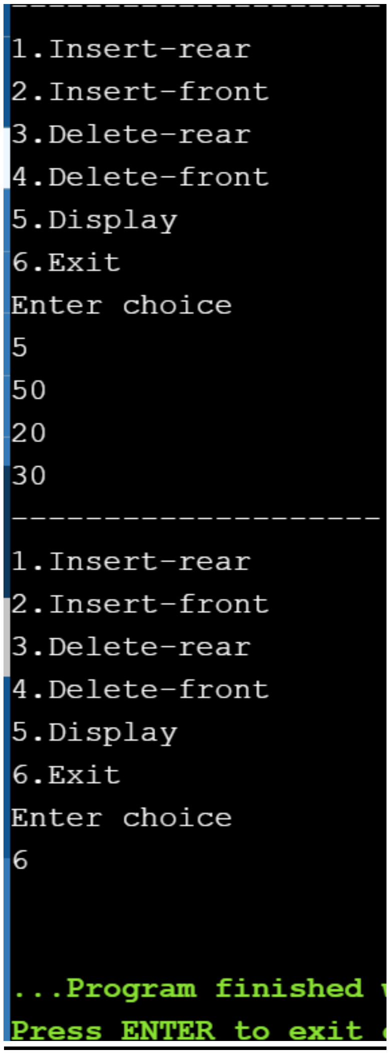
```
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
1
Enter the item
20
--------------------
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
1
Enter the item
30
--------------------
```



```
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
1
Enter the item
40
Queue overflow
--------------------
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
5
10
20
30
```



```
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
4
Item deleted is 10
--------------------
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
2
Enter the item
50
--------------------
```



```
--------------------
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
5
50
20
30
--------------------
1.Insert-rear
2.Insert-front
3.Delete-rear
4.Delete-front
5.Display
6.Exit
Enter choice
6


...Program finished
Press ENTER to exit
```