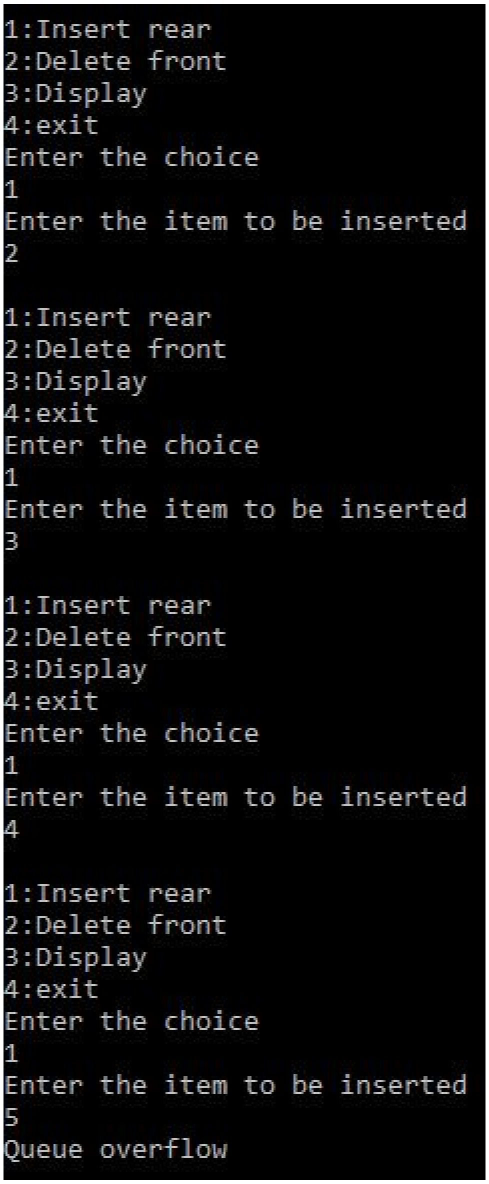
Write a Program to simulate the working of queue of integers using an array. Provide the following operations.
a) Insert Rear
b) Delete Front
c) Display the contents of queue
The program should print the appropriate messages for a queue empty and queue full condition.

CODE:

```
#include<stdio.h>
#include<process.h>
#define QUE_SIZE 3
int item,front=0,rear=-1,q[10];
void insertrear()
{

  if(rear==QUE_SIZE-1)
  {
    printf("Queue overflow\n");
    return ;
  }
  rear=rear+1;
  q[rear]=item;
}
int deletefront()
{

  if(front>rear)
  {
    front=0;
    rear=-1;
    return -1;
  }
  return q[front++];
}
void displayQ()
{
  int i;
  if(front>rear)
  {
    printf("Queue is empty\n");
    return ;
  }
  printf("Contents of queue\n");
  for(i=front;i<=rear;i++)
  {
    printf("%d\n",q[i]);
  }
}
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=%d\n",item);
            break;
            case 3:displayQ();
            break;
            default:exit(0);
        }
    }
}
```

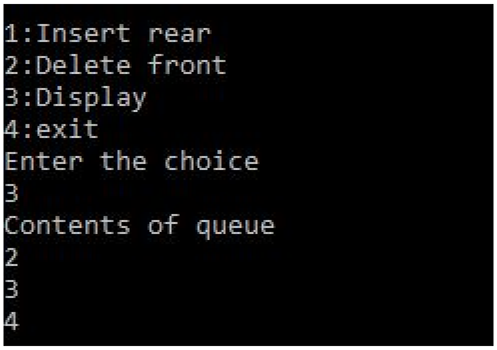
OUTPUT: Including all operation(insertrear,deletefront,display,exit) and overflow,queue empty conditions.

(Inserting items and overflow condition)



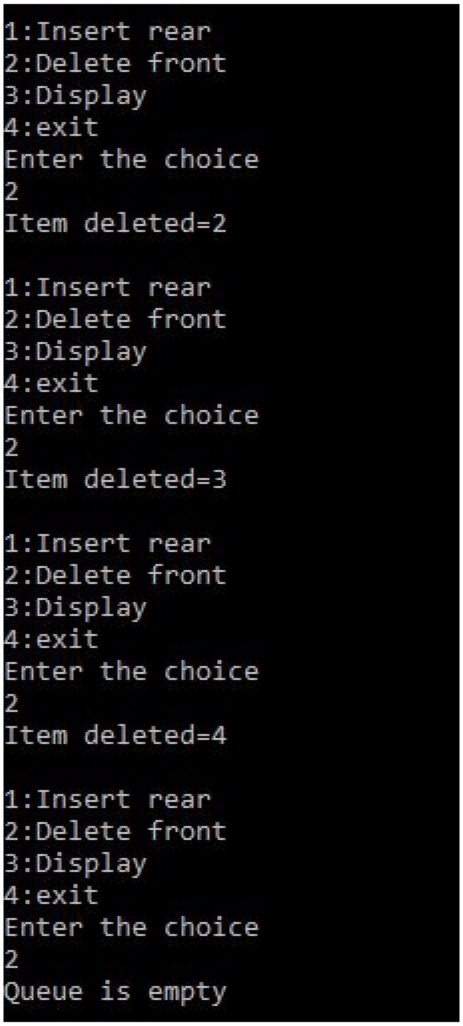
```
1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
1
Enter the item to be inserted
2

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

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

1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
1
Enter the item to be inserted
5
Queue overflow
```

(Display operation)



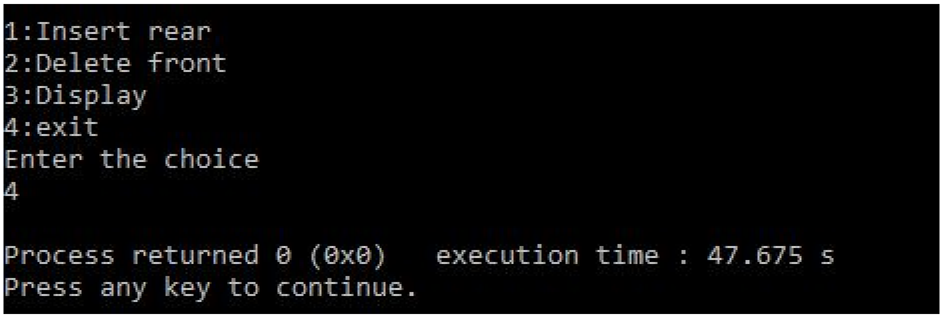
```
1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
3
Contents of queue
2
3
4
```

(Deleting items and queue empty condition)



```
1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
2
Item deleted=2

1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
2
Item deleted=3

1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
2
Item deleted=4

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

(Exit operation)



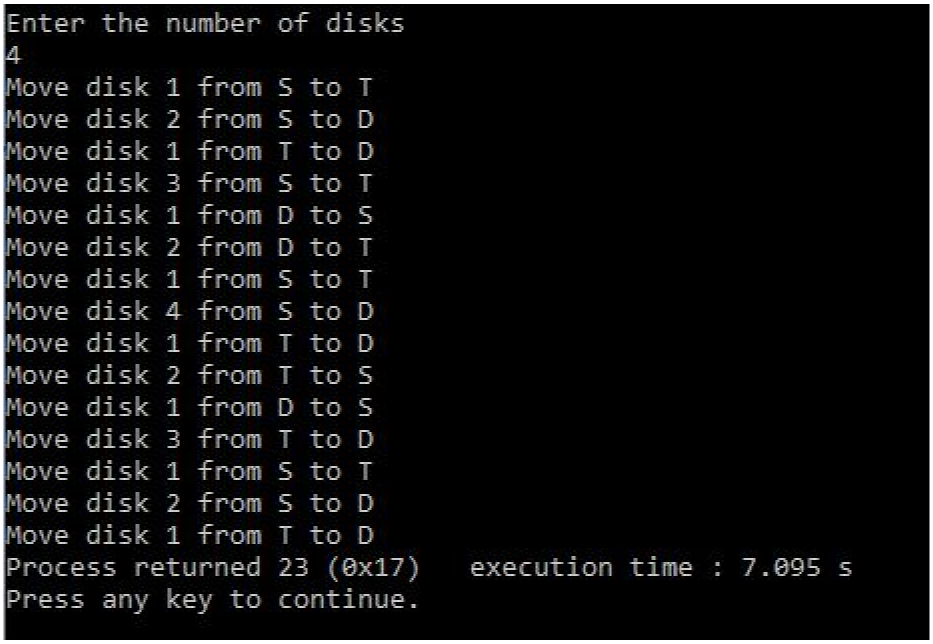
```
1:Insert rear
2:Delete front
3:Display
4:exit
Enter the choice
4

Process returned 0 (0x0)   execution time : 47.675 s
Press any key to continue.
```

Tower of Hanoi

CODE:

```
#include<stdio.h>
void towers(int n,char src,char temp,char dest)
{
  if(n==1)
  {
    printf("Move disk 1 from %c to %c",src,dest);
    return ;
  }
  towers(n-1,src,dest,temp);
  printf("\nMove disk %d from %c to %c\n",n,src,dest);
  towers(n-1,temp,src,dest);
}
void main()
{
  int n;
  printf("Enter the number of disks\n");
  scanf("%d",&n);
  towers(n,'S','T','D');
}
```

OUTPUT:



```
Enter the number of disks
4
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
Move disk 3 from S to T
Move disk 1 from D to S
Move disk 2 from D to T
Move disk 1 from S to T
Move disk 4 from S to D
Move disk 1 from T to D
Move disk 2 from T to S
Move disk 1 from D to S
Move disk 3 from T to D
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
Process returned 23 (0x17)   execution time : 7.095 s
Press any key to continue.
```