Circular Queue

```
#include<stdio.h>
#include<conio.h>
#include<process.h>  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\n");
    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\n");
    return;
  }
f=front;
printf("Contents of queue\n");
for(int i=1; i<=count; i++)
  {
    printf("%d\n", q[f]);
    f = (f+1)%QUE_SIZE;
}}
```

```
void main ()
{
  int choice;
  for(;;)
  {
    printf ("\n 1: insertrear \n2: deletefront \n 3: display
          \n 4: 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: display();
            break;

    case 4: exit (0);
     break;
     default: printf ("Invalid choice \n");
    }
  }
}
```