CIRCULAR QUEUE.

```
#include<stdio.h>
#include<conio.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)
     {
       pf (" 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;
   vo }
    void display()
     {
      int i, f;
      if (count ==0)
       {
       pf (" Queue is empty\n");
return;
 }
 f=front;
pf("contents of queue\n");
  for(i=1;i<=count;i++)
  {
  pf(" %d\n", q[f]);
  f=(f+1) % QUE_SIZE;
  }
}
void main()
  {
   int choice;
  for(;;)
 {
pf(" 1:insert rear 2:deletefront 3: display 4:exit");
 pf("Enter the choice");
  sf(" %d", &choice);
 switch(choice).
 {
 case 1: printf("enter the item to be inserted\n");
 sf("%d", &item);
   insertrear();
    break;
 case 2:  item=deletefront();
      if (item==-1);
    pf(" queue is empty\n");
      else
   pf(" item deleted =%d\n", item);
      break.
   case 3: displayQ();
          break;
   default: exit(0);
  }
} }
```