```
/*6. Design, Develop and Implement a menu driven Program in C for the
following operations on
      Circular QUEUE of Characters(Array Implementation of Queue with
maximum size MAX)
            a.Insert an Element on to Circular QUEUE
            b.Delete an Element from Circular QUEUE
            c.Demonstrate Overflow and Underflow situations on Circular
QUEUE
            d.Display the status of Circular QUEUE
            e.Exit
      Support the program with appropriate functions for each of the
above operations*/

#define MAX 5
#include<math.h>
#include<stdio.h>
#include<stdlib.h>

int cQueue[MAX], front= 0, rear= 0;

int cQFull(void)                    /* Function for Overflow operation */
{
      if ((rear+1)%MAX == front )
      {
        printf("Circular Queue is FULL\n");
        return 1;
      }
      else
            return 0;
}

int cQEmpty(void)                          /* Function for underflow
operation */
{
      if ((front == rear))          /*if front and rear both are same */
    {
        printf("Circular Queue is Empty\n");
        return 1;
    }
      else
            return 0;
}

void cqinsert(int ele)              /* Function for cqinsert operation */
{
      if (!cQFull())
      {
            rear = (rear+1) % MAX;
            cQueue[rear] = ele;
      }
}


int cqdelete(void)                  /* Function for cqdelete operation */
{
    if (!cQEmpty())
      {
            front = (front+1) % MAX;
```

```
            return cQueue[front];
    }
}


void display(void)           /* Function for display operation */
{
      int i;
      if (!cQEmpty())

      {
          i = front+1;
            printf("\nCircular Queue Elements are : \n");
            while(i != rear)
        {
            printf("%d\t", cQueue[i]);
            i=(i+1) % MAX;
        }
        printf("%d\n", cQueue[i]);
      }
}

int main()
{
      int  ele, ch;
      while (1)
      {
            printf("\n********Circular Queue Operations Menu********\n");
            printf("1. Insert Operation\n");
            printf("2. Delete Operation\n");
            printf("3. Display Circular Queue\n");
            printf("4. Exit\n");
            printf("Enter your choice:\n");
            scanf("%d", &ch);
            switch (ch)
            {
            case 1: printf("Enter an element to insert\n");
                  scanf("%d", &ele);
                  cqinsert(ele);
                  break;
            case 2: ele = cqdelete();
                  if(ele!=-1)
                  printf("Deleted element is:  %d\n", ele);
                  break;
            case 3: display();
                  break;
            case 4: exit(0);
            default: printf("Enter the valid choice\n\n");
                  break;
            }
      }
      return 0;
}
```