

Circular Queue

```
#include<stdio.h>
#include<stdlib.h>

#define MAX 5

char circular_queue[MAX];
int front = -1, rear = -1;

int isEmpty()
{
    if (front == -1 && rear == -1)
        return 1;
    else
        return 0;
}

int isFull()
{
    if ((rear + 1) % MAX == front)
        return 1;
    else
        return 0;
}

void insertElement(char element)
{
    if (isFull())
    {
        printf("Circular Queue Overflow\n");
        return;
    }
    else if (isEmpty())
    {
        front = rear = 0;
    }
    else
    {
        rear = (rear + 1) % MAX;
    }
    circular_queue[rear] = element;
}

void deleteElement()
```

```
{  
    if (isEmpty())  
    {  
        printf("Circular Queue Underflow\n");  
        return;  
    }  
    else if (front == rear)  
    {  
        front = rear = -1;  
    }  
    else  
    {  
        front = (front + 1) % MAX;  
    }  
}  
  
void display()  
{  
    int i;  
    if (isEmpty())  
    {  
        printf("Circular Queue is empty\n");  
        return;  
    }  
    printf("Circular Queue elements: ");  
    i = front;  
    do  
    {  
        printf("%c ", circular_queue[i]);  
        i = (i + 1) % MAX;  
    }  
    while (i != (rear + 1) % MAX);  
    printf("\n");  
}  
  
int main()  
{  
    int choice;  
    char element;  
    do  
    {  
        printf("\n\n--- Circular Queue Menu ---\n");  
        printf("1. Insert an Element\n");  
        printf("2. Delete an Element\n");  
        printf("3. Display Circular Queue\n");
```

```
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

switch(choice)
{
case 1:
    printf("Enter element to be inserted: ");
    scanf(" %c", &element);
    insertElement(element);
    break;
case 2:
    deleteElement();
    break;
case 3:
    display();
    break;
case 4:
    printf("Exiting...\n");
    break;
default:
    printf("Invalid choice! Please enter a valid option.\n");
}
}

while(choice != 4);

return 0;
}
```