circularqueue.c

```c
#include<stdio.h>
#include<stdlib.h>
#define QSIZE 4

int q[QSIZE], r = -1, f = 0, count = 0, item;

/* Insert Operation */
void insert() {
    /* Check for queue overflow */
    if (count == QSIZE) {
        printf("Queue Overflow\n");
        return;
    }
    r = (r + 1) % QSIZE; /* Increment rear by 1 */
    q[r] = item; /* Insert into queue */
    count++;
}

/* Delete Operation */
void del() {
    /* Check for Queue Underflow */
    if (count == 0) {
        printf("Queue Underflow\n");
        return;
    }
    printf("The item deleted is: %d\n", q[f]);
    f = (f + 1) % QSIZE;
    count--;
}

/* Display Operation */
void display(int front) {
    int i;
    /* Check for Empty Queue */
    if (count == 0) {
        printf("Queue is Empty\n");
        return;
```

```
    }
    /* Display the contents of the queue */
    printf("Contents of the queue:\n");
    for (i = 1; i <= count; i++) {
        printf("%d\n", q[front]);
        front = (front + 1) % QSIZE;
    }
}

void main() {
    int choice;
    do {
        printf("************************\n");
        printf("Circular Queue Operations\n");
        printf("1. Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Quit\n");
        printf("Enter your choice:\n");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter the item to be inserted:\n");
                scanf("%d", &item);
                insert();
                break;
            case 2:
                del();
                break;
            case 3:
                display(f);
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice\n");
        }
    } while (choice != 4);
}
```