```c
#include <stdio.h>
#define MAX 5

int cq[MAX];
int front = -1, rear = -1;

void insert() {
    int item;

    if ((front == 0 && rear == MAX - 1) || (front == rear + 1)) {
        printf("Circular Queue Overflow!\n");
        return;
    }

    printf("Enter element to insert: ");
    scanf("%d", &item);

    if (front == -1) {
        front = rear = 0;
    } else {
        rear = (rear + 1) % MAX;
    }

    cq[rear] = item;
    printf("Inserted %d\n", item);
}

void delete() {
    if (front == -1) {
        printf("Circular Queue Underflow!\n");
        return;
    }
```

=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

```c
            printf("Deleted %d\n", cq[front]);

        if (front == rear) {

            front = rear = -1;
        } else {
            front = (front + 1) % MAX;
        }
    }
}

void display() {
    if (front == -1) {
        printf("Circular Queue is Empty!\n");
        return;
    }

    printf("Circular Queue elements: ");

    int i = front;
    while (1) {
        printf("%d ", cq[i]);
        if (i == rear)
            break;
        i = (i + 1) % MAX;
    }
    printf("\n");
}

int main() {
    int choice;

    while (1) {
        printf("\n--- Circular Queue Menu ---\n");
```

```c
        }

        printf("Circular Queue elements: ");

        int i = front;
        while (1) {
            printf("%d ", cq[i]);
            if (i == rear)
                break;
            i = (i + 1) % MAX;
        }
        printf("\n");
    }

int main() {
    int choice;

    while (1) {
        printf("\n--- Circular Queue Menu ---\n");
        printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: insert(); break;
            case 2: delete(); break;
            case 3: display(); break;
            case 4: return 0;
            default: printf("Invalid Choice!\n");
        }
    }
}
```

```
--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Circular Queue is Empty!

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 6
Inserted 6

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 9
Inserted 9

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 7
Inserted 7

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 8
Inserted 8

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 10
Inserted 10

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Circular Queue Overflow!

--- Circular Queue Menu ---
```
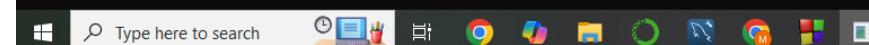
```
--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Circular Queue Overflow!

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 11
Invalid Choice!

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Circular Queue elements: 6 9 7 8 10

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted 6

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4

Process returned 0 (0x0)   execution time : 53.316 s
Press any key to continue.
```