PRACTICAL NO: 3

Name-Saloni Vishwakarma

Batch-Roll no: C1-13

Aim: To study and implement Circular Queue ADT.

Code:

```
#include <stdio.h>
#include <conio.h>
#define MAX 10
int queue[MAX];
int front = -1,
rear = -1;
void insert(void);
int delete_element(void);
int peek(void);
void display(void);
int main() {
   int option, val;
   do {
      printf("\n ***** MAIN MENU *****");
      printf("\n 1. Insert an element");
      printf("\n 2. Delete an element");
      printf("\n 3. Peek");
      printf("\n 4. Display the queue");
      printf("\n 5. EXIT");
      printf("\n Enter your option : ");
      scanf("%d", & option);
```

```
        switch (option) {
            case 1:
                insert();
                break;
            case 2:
                val = delete_element();
                if (val != -1)
                printf("\n The number deleted is : %d", val);
                break;
            case 3:
                val = peek();
                if (val != -1)
                printf("\n The first value in queue is : %d", val);
                break;
            case 4:
                display();
                break;
        }
    }while (option != 5);
    getch();
    return 0;
}
void insert() {
    int num;
    printf("\n Enter the number to be inserted in the queue : ");
    scanf("%d", & num);
    if (front == 0 && rear == MAX - 1)
```

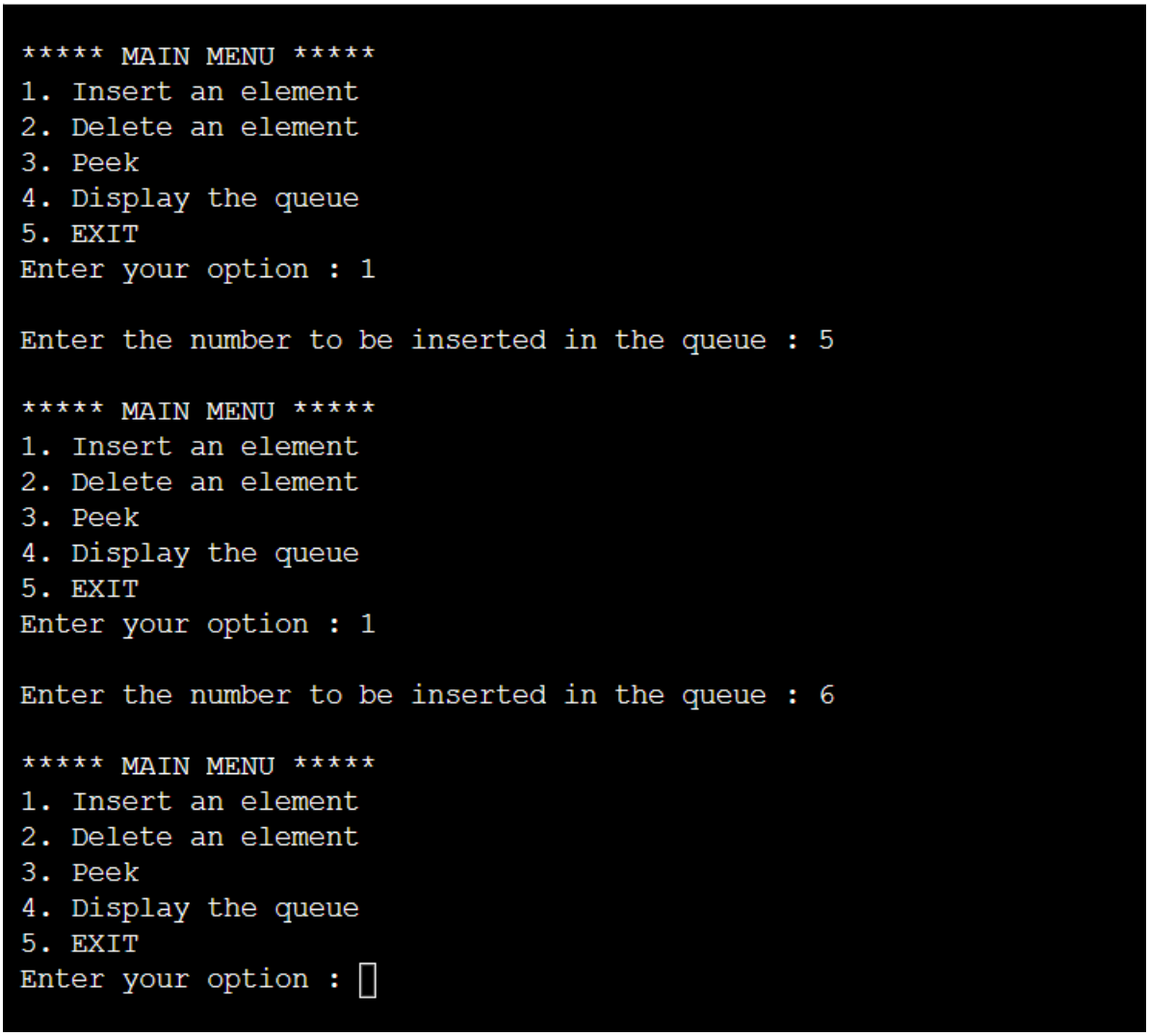
```
    printf("\n OVERFLOW");
    else if (front == -1 && rear == -1) {
        front = rear = 0;
        queue[rear] = num;
    }
    else if (rear == MAX - 1 && front != 0) {
        rear = 0;
        queue[rear] = num;
    }
    else {
        rear++;
        queue[rear] = num;
        }
}
int delete_element() {
    int val;
    if (front == -1 && rear == -1) {
        printf("\n UNDERFLOW");
        return -1;
    }
    val = queue[front];
    if (front == rear) front = rear = -1;
    else {
        if (front == MAX - 1)
        front = 0;
        else front++;
        }
```

```
    return val;
}
int peek() {
    if (front == -1 && rear == -1) {
        printf("\n QUEUE IS EMPTY");
        return -1;
    }
    else {
        return queue[front];
    }
}
void display() {
    int i;
    printf("\n");
    if (front == -1 && rear == -1)
    printf("\n QUEUE IS EMPTY");
    else {
        if (front < rear) {
            for (i = front; i <= rear; i++)
            printf("\t %d", queue[i]);
        }
        else {
            for (i = front; i < MAX; i++)
            printf("\t %d", queue[i]);
            for (i = 0; i <= rear; i++)
            printf("\t %d", queue[i]);
        }
```

```
    }
}
```

Output:

1. Insert



2. Delete

```
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 1

Enter the number to be inserted in the queue : 4

***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 2

The number deleted is : 4
```

3. Peek

```
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 1

Enter the number to be inserted in the queue : 9

***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 3

The first value in queue is : 9
```

4. Display

```
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 1

Enter the number to be inserted in the queue : 4

***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 1

Enter the number to be inserted in the queue : 5

***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 4

        4       5
```