Experiment 3

CODE:

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

#define MAX 5

typedef struct {
    int items[MAX];
    int front, rear;
} Queue;

void initQueue(Queue *q) {
    q->front = -1;
    q->rear = -1;
}

int isQueueEmpty(Queue *q) {
    return q->front == -1;
}

int isQueueFull(Queue *q) {
    return (q->rear + 1) % MAX == q->front;
}

void enqueue(Queue *q, int item) {
    if (isQueueFull(q)) {
        printf("Queue is full. Cannot enqueue %d.\n", item);
        return;
    }
    if (isQueueEmpty(q)) {
        q->front = 0;
    }
    q->rear = (q->rear + 1) % MAX;
    q->items[q->rear] = item;
    printf("Enqueued %d\n", item);
}

int dequeue(Queue *q) {
    if (isQueueEmpty(q)) {
        printf("Queue is empty. Cannot dequeue.\n");
        return -1;
    }
    int item = q->items[q->front];
    if (q->front == q->rear) {
        initQueue(q);
    } else {
        q->front = (q->front + 1) % MAX;
    }
    return item;
}

int getFront(Queue *q) {
    if (isQueueEmpty(q)) {
        printf("Queue is empty.\n");
        return -1;
    }
    return q->items[q->front];
}

int getRear(Queue *q) {
    if (isQueueEmpty(q)) {
        printf("Queue is empty.\n");
        return -1;
    }
    return q->items[q->rear];
}

```

```c
void displayQueue(Queue *q) {
    if (isQueueEmpty(q)) {
        printf("Queue is empty.\n");
        return;
    }
    printf("Queue contents: ");
    int i = q->front;
    while (1) {
        printf("%d ", q->items[i]);
        if (i == q->rear) break;
        i = (i + 1) % MAX;
    }
    printf("\n");
}

int main() {
    Queue q;
    initQueue(&q);
    int choice, item;

    do {
        printf("\nQueue Menu:\n");
        printf("1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Get Front\n");
        printf("4. Get Rear\n");
        printf("5. Check if Queue is Empty\n");
        printf("6. Check if Queue is Full\n");
        printf("7. Display Queue\n");
        printf("8. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter the item to enqueue: ");
                scanf("%d", &item);
                enqueue(&q, item);
                displayQueue(&q);
                break;
            case 2:
                item = dequeue(&q);
                if (item != -1) {
                    printf("Dequeued %d\n", item);
                }
                displayQueue(&q);
                break;
            case 3:
                item = getFront(&q);
                if (item != -1) {
                    printf("Front item is %d\n", item);
                }
                break;
            case 4:
                item = getRear(&q);
                if (item != -1) {
                    printf("Rear item is %d\n", item);
                }
                break;
            case 5:
                if (isQueueEmpty(&q)) {
                    printf("Queue is empty.\n");
                } else {
                    printf("Queue is not empty.\n");
                }
                break;
            case 6:
                if (isQueueFull(&q)) {
                    printf("Queue is full.\n");
                } else {
                    printf("Queue is not full.\n");
                }
                break;
```

```
140         case 7:
141             displayQueue(&q);
142             break;
143         case 8:
144             printf("Exiting...\n");
145             break;
146         default:
147             printf("Invalid choice! Please try again.\n");
148             break;
149     }
150     } while (choice != 8);
151
152     return 0;
153 }
```

OUTPUT:

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 1
Enter the item to enqueue: 85
Enqueued 85
Queue contents: 85

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 1
Enter the item to enqueue: 58
Enqueued 58
Queue contents: 85 58

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 1
Enter the item to enqueue: 98
Enqueued 98
Queue contents: 85 58 98

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue

8. Exit
Enter your choice: 2
Dequeued 85
Queue contents: 58 98

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 3
Front item is 58

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 4
Rear item is 98

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 6
Queue is not full.

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 2
Dequeued 58
Queue contents: 98

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 2
Dequeued 98
Queue is empty.

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear

5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 5
Queue is empty.

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 1
Enter the item to enqueue: 85
Enqueued 85
Queue contents: 85

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 1
Enter the item to enqueue: 89
Enqueued 89
Queue contents: 85 89

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 7
Queue contents: 85 89 75

Queue Menu:
1. Enqueue
2. Dequeue
3. Get Front
4. Get Rear
5. Check if Queue is Empty
6. Check if Queue is Full
7. Display Queue
8. Exit
Enter your choice: 8
Exiting...

=== Code Execution Successful ===