Practical 8c

Pra8C : Implement a Circular Queue and perform the Queue operations: Enqueue , Dequeue and Print using Menu Driver Program such as 1.Add, 2.Delete and 3. Print and 4. Exit.

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

#define MAX 5  // Define the maximum size of the queue

// Define the structure for Circular Queue
struct Queue {
   int items[MAX];
   int front;
   int rear;
};

// Function to initialize the queue
void initializeQueue(struct Queue* q) {
   q->front = 0;
   q->rear = 0;
}

// Function to check if the queue is empty
int isEmpty(struct Queue* q) {
   return q->front == q->rear;
}

// Function to check if the queue is full
int isFull(struct Queue* q) {
   return (q->rear + 1) % MAX == q->front;
}

// Function to enqueue (add) an item to the queue
```

```
void enqueue(struct Queue* q, int value) {
    if (isFull(q)) {
        printf("Queue is full. Cannot enqueue %d\n", value);
        return;
    }
    q->items[q->rear] = value;
    q->rear = (q->rear + 1) % MAX;  // Move rear to next position, wrap around if necessary
    printf("Enqueued %d to the queue\n", value);
}

// Function to dequeue (remove) an item from the queue
void dequeue(struct Queue* q) {
    if (isEmpty(q)) {
        printf("Queue is empty. Cannot dequeue\n");
        return;
    }
    int removedItem = q->items[q->front];
    q->front = (q->front + 1) % MAX;  // Move front to next position, wrap around if necessary
    printf("Dequeued %d from the queue\n", removedItem);
}

// Function to print the queue contents
void printQueue(struct Queue* q) {
    if (isEmpty(q)) {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue contents: ");
    int i = q->front;
    while (i != q->rear) {
        printf("%d ", q->items[i]);
        i = (i + 1) % MAX;  // Move to the next position in the circular queue
    }
```

```
    printf("\n");
}

// Menu function to drive the program
void menu() {
    struct Queue q;
    initializeQueue(&q);  // Initialize the queue
    int choice, value;

    while (1) {
        // Display the menu options
        printf("\nMenu:\n");
        printf("1. Add (Enqueue)\n");
        printf("2. Delete (Dequeue)\n");
        printf("3. Print Queue\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                // Enqueue operation
                printf("Enter the value to enqueue: ");
                scanf("%d", &value);
                enqueue(&q, value);
                break;

            case 2:
                // Dequeue operation
                dequeue(&q);
                break;

            case 3:
```

```
            // Print the queue
            printQueue(&q);
            break;

        case 4:
            // Exit the program
            printf("Exiting the program.\n");
            exit(0);

        default:
            printf("Invalid choice. Please enter a valid option.\n");
      }
   }
}

int main() {
   menu();  // Start the menu-driven program
   return 0;
}
```