CSA0317-DATA STRUCTURES

Program 13

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
int queue[MAX];
int front = -1, rear = -1;
void enqueue(int value) {
    if (rear == MAX - 1) {
        printf("Queue Overflow! Cannot enqueue %d\n", value);
    } else {
        if (front == -1) {
            front = 0;
        }
        rear++;
        queue[rear] = value;
        printf("%d enqueued to queue\n", value);
    }
}
void dequeue() {
    if (front == -1 || front > rear) {
        printf("Queue Underflow! Queue is empty\n");
    } else {
        printf("%d dequeued from queue\n", queue[front]);
        front++;
```

```c
        if (front > rear) {
            front = rear = -1;
        }
    }
}
void display() {
    if (front == -1) {
        printf("Queue is empty\n");
    } else {
        printf("Queue elements: ");
        for (int i = front; i <= rear; i++) {
            printf("%d ", queue[i]);
        }
        printf("\n");
    }
}
int main() {
    int choice, value;
    while (1) {
        printf("\nQueue Operations:\n");
        printf("1. ENQUEUE\n");
        printf("2. DEQUEUE\n");
        printf("3. DISPLAY\n");
        printf("4. EXIT\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
```

```c
    switch (choice) {
        case 1:
            printf("Enter value to enqueue: ");
            scanf("%d", &value);
            enqueue(value);
            break;
        case 2:
            dequeue();
            break;
        case 3:
            display();
            break;
        case 4:
            exit(0);
        default:
            printf("Invalid choice! Please try again.\n");
    }
}
return 0;
}
```

Output:

```
Queue Operations:
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
4. EXIT
Enter your choice: 1
Enter value to enqueue: 12
12 enqueued to queue

Queue Operations:
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
4. EXIT
Enter your choice: 1
Enter value to enqueue: 34
34 enqueued to queue

Queue Operations:
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
4. EXIT
Enter your choice: 2
12 dequeued from queue

Queue Operations:
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
4. EXIT
Enter your choice: 3
Queue elements: 34

Queue Operations:
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
4. EXIT
Enter your choice: 4


=== Code Execution Successful ===
```