Rajalakshmi Engineering College

Name: Parvendhan C
Email: 241801197@rajalakshmi.edu.in
Roll no: 241801197
Phone: 8270861183
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_PAH

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Guide Harish in developing a simple queue system for a customer service center. The customer service center can handle up to 25 customers at a time. The queue needs to support basic operations such as adding a customer to the queue, serving a customer (removing them from the queue), and displaying the current queue of customers.

Use an array for implementation.

Input Format

The first line of the input consists of an integer N, the number of customers arriving at the service center.

The second line consists of N space-separated integers, representing the customer IDs in the order they arrive.

Output Format

After serving the first customer in the queue, display the remaining customers in the queue.

If a dequeue operation is attempted on an empty queue, display "Underflow".

If the queue is empty, display "Queue is empty".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
101 102 103 104 105
Output: 102 103 104 105

Answer

```
#include <stdio.h>
#define MAX_SIZE 25
int queue[MAX_SIZE];
int front = -1;
int rear = -1;
void enqueue(int customerID) {
    if (rear == MAX_SIZE - 1) {
        printf("Queue is full\n");
        return;
    }
    if (front == -1) {
        front = 0;
    }
    rear++;
    queue[rear] = customerID;
}
void dequeue() {
    if (front == -1) {
        printf("Underflow\n");
        return;
    }
    front++;
```

```
    if (front > rear) {
        front = rear = -1;
        printf("Queue is empty\n");
    }
}
void displayQueue() {
    if (front == -1) {
        printf("Queue is empty\n");
        return;
    }
    for (int i = front; i <= rear; i++) {
        if (i != front) printf(" ");
        printf("%d", queue[i]);
    }
    printf("\n");
}
int main() {
    int N;
    scanf("%d", &N);
    if (N == 0) {
        printf("Underflow\nQueue is empty\n");
        return 0;
    }
    for (int i = 0; i < N; i++) {
        int customerID;
        scanf("%d", &customerID);
        enqueue(customerID);
    }
    dequeue();
    displayQueue();
    return 0;
}
```

Status : Correct **Marks : 10/10**

2. Problem Statement

Amar is working on a project where he needs to implement a special type of queue that allows selective dequeuing based on a given multiple. He wants to efficiently manage a queue of integers such that only elements not divisible by a given multiple are retained in the queue after a selective