Real Time Problem

BASIC BANKING SYSTEM

Introduction:

The basic banking system project simulates a simplified scenario of a bank's queue management system. It allows customers to join a queue and get served in a first-come-first-served manner.

Queue Data Sturcture:

- Implemented a circular queue data structure using an array to manage customers waiting in line.
- Utilized front and rear pointers to keep track of the front and rear ends of the queue.
- Defined functions like initQueue(), isEmpty(), isFull(), enqueue(), dequeue(), and displayQueue() to manage the queue operations efficiently.

User Interaction:

- Customers can choose to join the queue by entering their customer number.
- They are informed if the queue is full and prompted to wait if necessary.
- The system serves the next customer in line and displays their customer number.
- Customers can view the current queue to check their position.

Summary:

- The basic banking system project demonstrates the practical implementation of queue data structure concepts in a real-world scenario.

- The project lays the groundwork for more complex queue management systems and serves as an educational tool for understanding queue data structures and their applications.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_QUEUE_SIZE 10
int front = -1;
int rear = -1;
int size = 0;
int bankQueue[MAX_QUEUE_SIZE];
void initQueue() {
   front = -1;
   rear = -1;
   size = 0;
}
int isEmpty() {
   return (size == 0);
}

int isFull() {
   return (size == MAX_QUEUE_SIZE);
}
void enqueue(int value) {
   if (isFull()) {
      printf("Queue is full, cannot enqueue.\n");
      return;
   }
   if (isEmpty()) {
      front = rear = 0;
   } else {
      rear = (rear + 1) % MAX_QUEUE_SIZE;
   }
   bankQueue[rear] = value;
   size++;
}
int dequeue() {
   int item;
   if (isEmpty()) {
      printf("Queue is empty, cannot dequeue.\n");
      return -1;
   }
   item = bankQueue[front];
   if (front == rear) {
      front = rear = -1;
   } else {
      front = (front + 1) % MAX_QUEUE_SIZE;
   }
   size--;
   return item;
}
void displayQueue() {
   if (isEmpty()) {
      printf("Queue is empty.\n");
      return;
   }
   printf("Customers in queue: ");
   int i = front;
   int count = 0;
```

```
    while (count < size) {
        printf("%d, ", bankQueue[i]);
        i = (i + 1) % MAX_QUEUE_SIZE;
        count++;
    }
    printf("\n");
}

int main() {
    initQueue();

    int choice, customerNumber;
    do {
        printf("\nBanking System Menu:\n");
        printf("1. Join the queue\n");
        printf("2. Serve the next customer\n");
        printf("3. Display queue\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                if (!isFull()) {
                    printf("Enter your customer number: ");
                    scanf("%d", &customerNumber);
                    enqueue(customerNumber);
                    printf("Customer %d joined the queue.\n", customerNumber);
                } else {
                    printf("Queue is full. Please wait for some time.\n");
                }
                break;
            case 2:
                if (!isEmpty()) {
                    customerNumber = dequeue();
                    printf("Serving customer %d.\n", customerNumber);
                } else {
                    printf("Queue is empty. No customers to serve.\n");
                }
                break;
            case 3:
                displayQueue();
                break;
            case 4:
                printf("Exiting the banking system.\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 4);

    return 0;
}
```

Output:

Banking System Menu:

1. Join the queue
2. Serve the next customer
3. Display queue
4. Exit

Enter your choice: 1

Enter your customer number: 101

Customer 101 joined the queue.

Banking System Menu:

1. Join the queue
2. Serve the next customer
3. Display queue
4. Exit

Enter your choice: 2

Serving customer 101.

Banking System Menu:

1. Join the queue
2. Serve the next customer
3. Display queue
4. Exit

Enter your choice: 1

Enter your customer number: 102

Customer 102 joined the queue.

Banking System Menu:

1. Join the queue
2. Serve the next customer
3. Display queue
4. Exit

Enter your choice: 3

Customers in queue: 102

Banking System Menu:

1. Join the queue
2. Serve the next customer
3. Display queue
4. Exit

Enter your choice: 3

Customers in queue: 101, 102

Banking System Menu:

1. Join the queue
2. Serve the next customer
3. Display queue
4. Exit

Enter your choice: 4

Exiting the banking system.