BANK CUSTOMER QUEUE MANAGEMENT SYSTEM

INTRODUCTION

- Managing customer efficiently is a crucial task in banking operations.
- To simulate and practice queue management we developed a program that implements a customer queue system using the C programming language.
- The system uses a linked list that dynamically manage customers entering and leaving the queue.

OBJECTIVE

- To implement a queue data structure using linked lists in C.
- To manage th real-time addition (enqueue) and removal (dequeue) of customers.
- To display the queue status at any point .
- To understand dynamic memory management (malloc,free) in C.
- To simulate real-world queue handling in banks.

WHY C AND DSA

- C language is powerful ,close to hardware, and provides direct memory control,making it perfect for understanding how queues are managed internally.
- Data structures like linked list provide efficient ways to handle dynamic datasets without fixed memory size limitations.
- Algorithms help efficiently perform operations (enqueue,dequeue,display) in optimal time (O(1) for enqueue and dequeue).
- Learning C and DSA together strengthens the ability to solve real-world problems systematically.

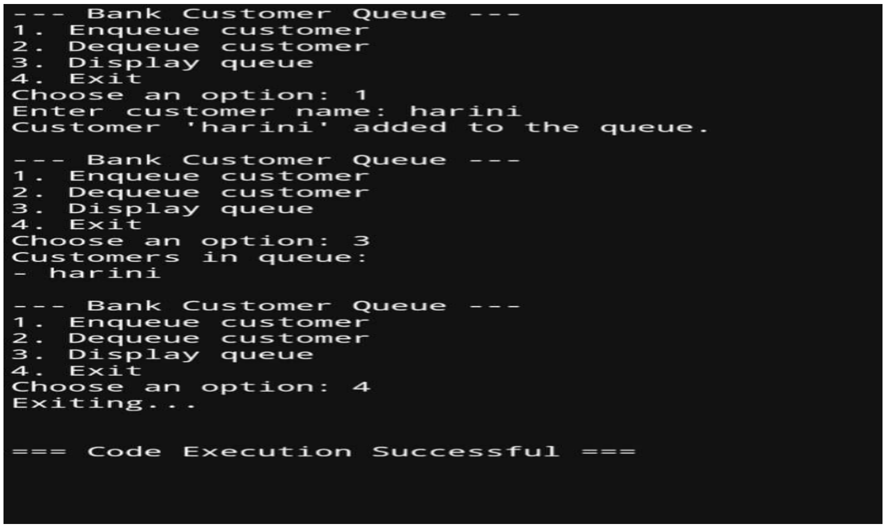
ALGORITHM

- 1.ENQUEUE (Adding a customer):
- Create a new node with customer's name.
- If the queue is empty then set front and rear to the new node.
- Else link the new node at the end and update rear.
- 2.DEQUEUE (serving the customer):
- Check if the queue is empty .if yes display a message.
- Else ,remove the front node and update front node to the next node.
- If after dequeue the queue becomes empty ,set rear to NULL.
- 3.DISPLAY QUEUE:
- Traverse from font to rear , printing each customer's name.

LESSONS LEARNT

- How to implement and manipulate a linked list.
- How queues world in real-world scenarios (FIFO: first in ,first out).
- Importance of dynamic memory allocation and avoiding memory leaks.
- Handling edge cases (like empty queue situations).
- Writing modular and readable code in C.
- Better understanding of algorithmic thinking and data management.

OUTPUT



```
--- Bank Customer Queue ---
1. Enqueue customer
2. Dequeue customer
3. Display queue
4. Exit
Choose an option: 1
Enter customer name: harini
Customer 'harini' added to the queue.

--- Bank Customer Queue ---
1. Enqueue customer
2. Dequeue customer
3. Display queue
4. Exit
Choose an option: 3
Customers in queue:
- harini

--- Bank Customer Queue ---
1. Enqueue customer
2. Dequeue customer
3. Display queue
4. Exit
Choose an option: 4
Exiting...


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

Thank you

Presented by:

S. Harini

G. Nandini

K. Rakshitha