🏦 Bank Teller Queue Simulation

This C program simulates customer arrivals and service at a bank with multiple tellers using a **Poisson process**. It models customer waiting times, service durations, and teller utilization over a typical **8-hour workday (480 minutes)**.

🚋 Overview

The simulation uses a queue to manage arriving customers and models the stochastic nature of arrivals using the **Poisson distribution**. Each customer is served by an available teller, with service times varying randomly between **2 and 3 minutes**.

At the end of the simulation, it provides a statistical summary of customer waiting times, including: - Mean (average) wait time
- Median wait time
- Mode wait time
- Standard deviation
- Longest wait time
- Total customers served

Features

- Simulates **random customer arrivals** using the Poisson process
- Handles **multiple tellers**
- Tracks and reports **waiting times** for all customers
- Dynamically manages the queue and memory allocation
- Computes **key statistical measures** for performance evaluation

---

Input Parameters

The program requires two inputs from the user:

1. **λ (lambda):** Average number of customers arriving per minute
2. **Number of tellers:** The number of service counters available

---

🏫 Output Example

After simulation, the program outputs a detailed report like this:

```
=== Simulation Report ===
Total customers served: 312
Average (Mean) wait time: 1.76 minutes
```

```
Median wait time: 1.00 minutes
Mode wait time: 0.00 minutes
Standard deviation: 1.54
Longest single wait: 7.00 minutes
```

Key Concepts Used

- **Poisson distribution:** Models the probability of a given number of arrivals per minute
- **Queue data structure:** Manages waiting customers in FIFO (first-in-first-out) order
- **Dynamic memory allocation:** Used to expand the array storing wait times
- **Statistical functions:** Mean, median, mode, and standard deviation calculations

Implementation Details

- **File:** `CPROJMCART.c`
- **Language:** C
- **Simulation duration:** 480 minutes (8 hours)
- **Service time per customer:** Randomly 2 or 3 minutes
- **Libraries used:**
- `<stdio.h>` for I/O
- `<stdlib.h>` for memory and random functions
- `<math.h>` for Poisson and statistical calculations
- `<time.h>` for random seeding

🚀 How to Run

1. Compile the program

```
gcc CPROJMCART.c -o simulation -lm
```

2. Run the executable

```
./simulation
```

3. Enter parameters when prompted

```
Enter the average number of customers arriving per minute (lambda): 0.5
Enter the number of tellers: 2
```

Memory Management

The program dynamically allocates and frees all memory used for: - Wait time array (automatically expands as needed) - Queue of customers
- Teller busy time tracker

No memory leaks occur if compiled and run properly.

🚤 Possible Extensions

You can enhance the simulation by: - Adding priority queues for VIP customers
- Logging customer data to a file
- Visualizing queue length and teller utilization over time
- Allowing variable service times (non-uniform distribution)

🏡 Author

Developed as part of a **simulation and data analysis project** to study queue behavior and statistical patterns in service systems.