Objectives

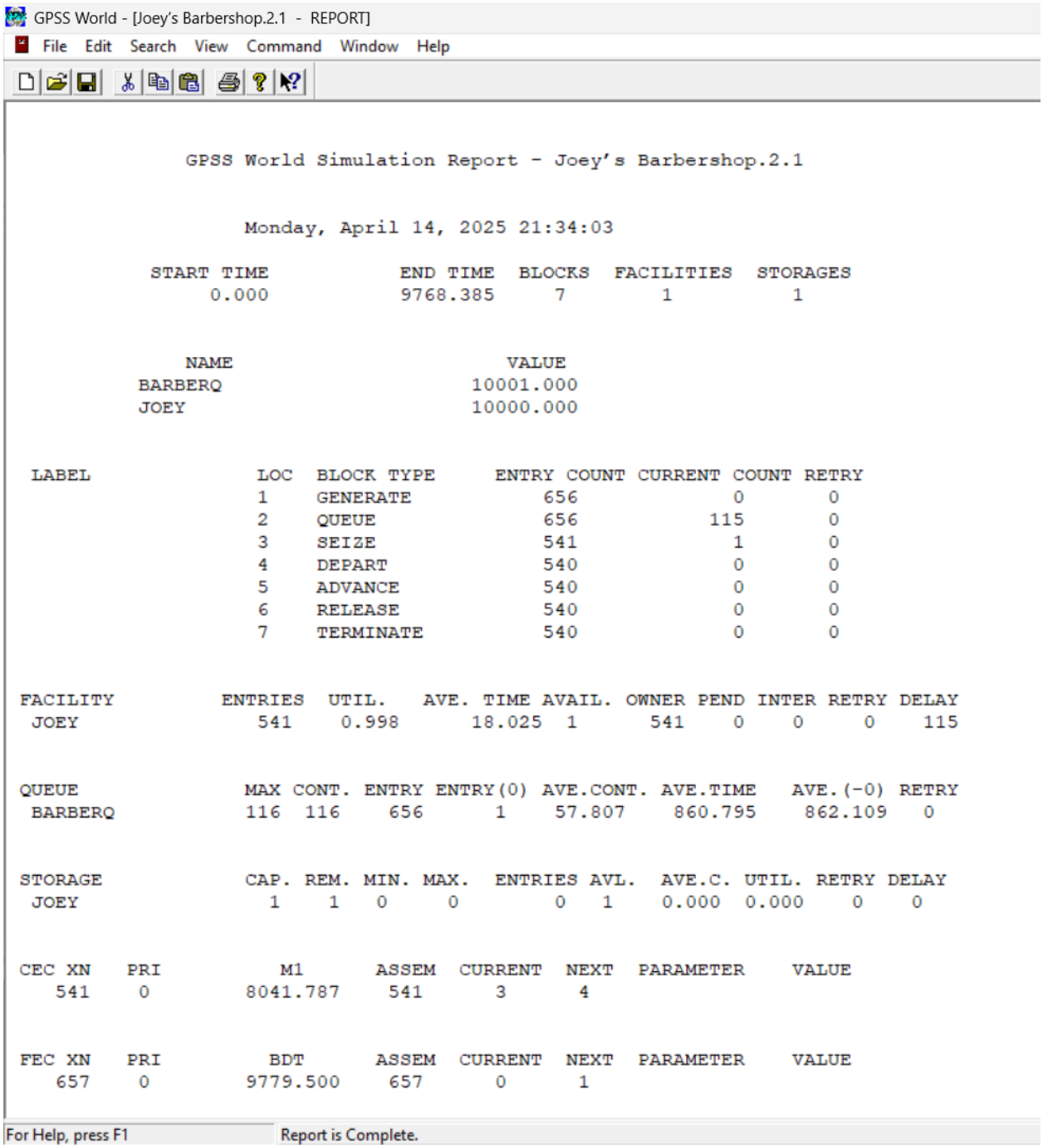
1. To understand and implement the concept of facilities in GPSS by simulating a barbershop scenario.
2. To model real-life situations using discrete event simulation techniques.
3. To apply statistical randomness (arrival and service time variations) in simulations.
4. To simulate the random walk (drunkard's walk) problem and understand probabilistic movement in one or two dimensions using any programming language.
5. To gain practical experience in using simulation as a tool for analyzing real-world systems.

Q.no.1. Customers arrive at Joey’s Barber Shop once every 15±3 minutes and it takes Joey 18±2 minutes to cut the hair of a customer. Write a GPSS program to simulate a barbershop using the concept of facility and run the simulation for 9 hours.

Source Code

```
GENERATE 15,3
QUEUE BARBERQ
SEIZE JOEY
DEPART BARBERQ
ADVANCE 18,2
RELEASE JOEY
TERMINATE 1
JOEY STORAGE 1
START 54
```

Output


GPSS World - [Joey's Barbershop.2.1 - REPORT]
File Edit Search View Command Window Help
GPSS World Simulation Report - Joey's Barbershop.2.1
Monday, April 14, 2025 21:34:03
START TIME END TIME BLOCKS FACILITIES STORAGES
0.000 9768.385 7 1 1
NAME VALUE
BARBERQ 10001.000
JOEY 10000.000
LABEL LOC BLOCK TYPE ENTRY COUNT CURRENT COUNT RETRY
1 GENERATE 656 0 0
2 QUEUE 656 115 0
3 SEIZE 541 1 0
4 DEPART 540 0 0
5 ADVANCE 540 0 0
6 RELEASE 540 0 0
7 TERMINATE 540 0 0
FACILITY ENTRIES UTIL. AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY
JOEY 541 0.998 18.025 1 541 0 0 0 115
QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY
BARBERQ 116 116 656 1 57.807 860.795 862.109 0
STORAGE CAP. REM. MIN. MAX. ENTRIES AVL. AVE.C. UTIL. RETRY DELAY
JOEY 1 1 0 0 0 1 0.000 0.000 0 0
CEC XN PRI M1 ASSEM CURRENT NEXT PARAMETER VALUE
541 0 8041.787 541 3 4
FEC XN PRI BDT ASSEM CURRENT NEXT PARAMETER VALUE
657 0 9779.500 657 0 1
For Help, press F1
Report is Complete.

Q.no.2. Simulate a random walk problem or a drunkard problem in any programming language.

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define STEPS 100

int main() {
    int x = 0, y = 0;
    srand(time(0));

    printf("Random Walk (2D) Starting from (0,0):\n");
    printf("Step 0: (%d, %d)\n", x, y);

    for (int i = 1; i <= STEPS; i++) {
        int direction = rand() % 4;

        switch (direction) {
            case 0: y++; break;       // North
            case 1: y--; break;       // South
            case 2: x++; break;       // East
            case 3: x--; break;       // West
        }

        printf("Step %d: (%d, %d)\n", i, x, y);
    }

    printf("Final Position after %d steps: (%d, %d)\n", STEPS, x, y);
    return 0;
}
```

.Output

```
Random Walk (2D) Starting from (0,0):
Step 0: (0, 0)
Step 1: (1, 0)
Step 2: (0, 0)
Step 3: (-1, 0)
Step 4: (-2, 0)
Step 5: (-2, 1)
Step 6: (-2, 0)
Step 7: (-2, 1)
Step 8: (-3, 1)
Step 9: (-2, 1)
Step 10: (-2, 0)
Step 11: (-1, 0)
Step 12: (0, 0)
Step 13: (0, -1)
Step 14: (0, 0)
Step 15: (-1, 0)
Step 16: (0, 0)
Step 17: (0, -1)
Step 18: (0, -2)
Step 19: (-1, -2)
Step 20: (0, -2)
Step 21: (0, -3)
Step 22: (-1, -3)
Step 23: (0, -3)
Step 24: (1, -3)
Step 25: (1, -2)
Step 26: (0, -2)
Step 27: (0, -1)
Step 28: (0, 0)
Step 29: (1, 0)
Step 30: (0, 0)
Step 31: (0, -1)
Step 32: (0, -2)
Step 33: (1, -2)
Step 34: (1, -1)
Step 35: (1, 0)
Step 36: (0, 0)
Step 37: (0, -1)
Step 38: (0, 0)
Step 39: (1, 0)
```

```
Step 63: (7, 0)
Step 64: (7, -1)
Step 65: (8, -1)
Step 66: (9, -1)
Step 67: (9, 0)
Step 68: (8, 0)
Step 69: (7, 0)
Step 70: (8, 0)
Step 71: (7, 0)
Step 72: (7, -1)
Step 73: (8, -1)
Step 74: (8, -2)
Step 75: (8, -1)
Step 76: (7, -1)
Step 77: (8, -1)
Step 78: (9, -1)
Step 79: (9, -2)
Step 80: (9, -3)
Step 81: (8, -3)
Step 82: (8, -4)
Step 83: (8, -5)
Step 84: (8, -4)
Step 85: (8, -3)
Step 86: (7, -3)
Step 87: (7, -4)
Step 88: (6, -4)
Step 89: (7, -4)
Step 90: (8, -4)
Step 91: (8, -3)
Step 92: (8, -4)
Step 93: (8, -3)
Step 94: (9, -3)
Step 95: (10, -3)
Step 96: (11, -3)
Step 97: (11, -2)
Step 98: (12, -2)
Step 99: (12, -3)
Step 100: (11, -3)
Final Position after 100 steps: (11, -3)
[1] + Done                       "/usr/bin/gdb" --in
>"/tmp/Microsoft-MIEngine-Out-io2g02nw.rh0"
➜  5th sem git:(main) ✗
```

Conclusion

In this lab, we successfully simulated Joey’s Barbershop using GPSS by incorporating randomness in customer arrival and service times. The use of facilities in GPSS allowed us to model the queueing and servicing process effectively. Additionally, by simulating the random walk problem, we gained insights into stochastic processes and their behavior over time. These simulations highlight the power of modeling and randomness in understanding complex systems and contribute to building foundational skills in simulation and probabilistic modeling.