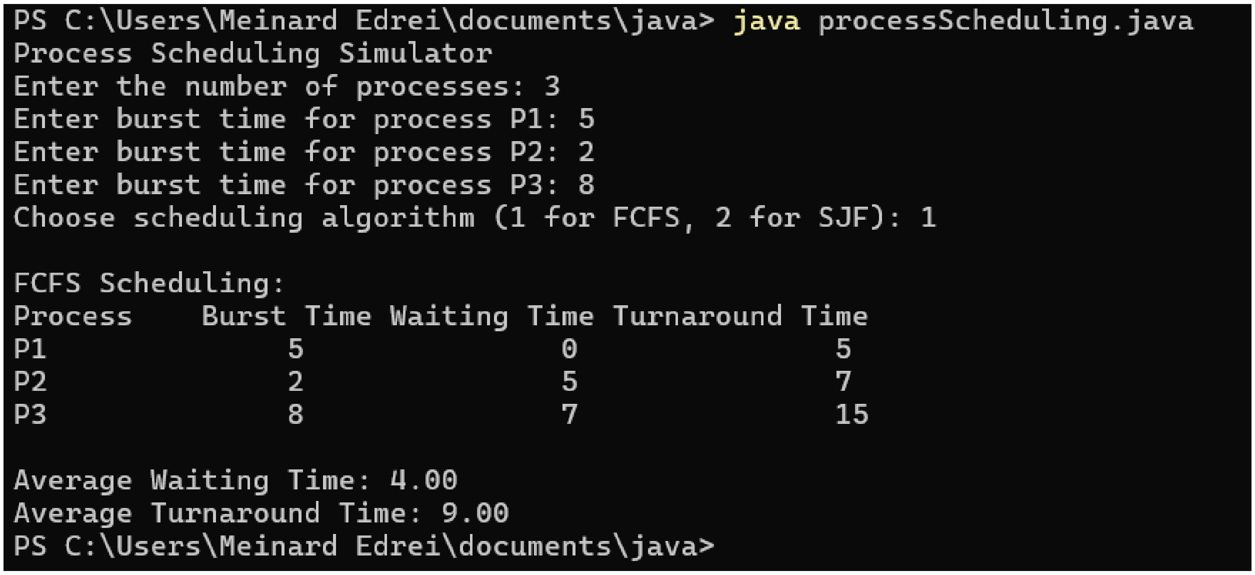
UNIVERSITY OF MAKATI
College of Computing and Information Sciences
OPERSYS
Week 7 Laboratory Activity

Scope: Process Scheduling Single Queueing (FCFS & SJF) - 50 points

<u>Members</u>:
Armojallas, Caleb Joshua
Canlas, Korbin
Manuel, John Lou
Santos, Meinard Edrei

General Instructions/Requirements:
1. Perform the given sample program for FCFS & SJF as Exercises.
2. Create a program that simulates FCFS and SJF depending on the user choice.

Output:



```
PS C:\Users\Meinard Edrei\documents\java> java processScheduling.java
Process Scheduling Simulator
Enter the number of processes: 3
Enter burst time for process P1: 5
Enter burst time for process P2: 2
Enter burst time for process P3: 8
Choose scheduling algorithm (1 for FCFS, 2 for SJF): 1

FCFS Scheduling:
Process    Burst Time Waiting Time Turnaround Time
P1              5               0               5
P2              2               5               7
P3              8               7               15

Average Waiting Time: 4.00
Average Turnaround Time: 9.00
PS C:\Users\Meinard Edrei\documents\java>
```

Code:

```
import java.util.Arrays;
import java.util.Comparator;
import java.util.Scanner;

public class ProcessScheduling {

    static void calculateFCFS(int[] burstTimes) {
        int n = burstTimes.length;
        int[] waitingTime = new int[n];
        int[] turnaroundTime = new int[n];
        // Calculate waiting time
        for (int i = 1; i < n; i++) {
            waitingTime[i] = waitingTime[i - 1] + burstTimes[i - 1];
        }

        // Calculate turnaround time
        for (int i = 0; i < n; i++) {
            turnaroundTime[i] = waitingTime[i] + burstTimes[i];
        }

        // Print results
        System.out.println("\nFCFS Scheduling:");
        System.out.printf("%-10s %-10s %-10s %-10s\n", "Process", "Burst Time", "Waiting
Time", "Turnaround Time");
        int totalWaitingTime = 0;
        int totalTurnaroundTime = 0;
        for (int i = 0; i < n; i++) {
            totalWaitingTime += waitingTime[i];
            totalTurnaroundTime += turnaroundTime[i];
            System.out.printf("P%d\t\t%d\t\t%d\t\t%d\n", i + 1, burstTimes[i], waitingTime[i],
turnaroundTime[i]);
        }

        System.out.printf("\nAverage Waiting Time: %.2f\n", (double) totalWaitingTime / n);
        System.out.printf("Average Turnaround Time: %.2f\n", (double) totalTurnaroundTime / n);
    }

    static void calculateSJF(int[] burstTimes) {
        int n = burstTimes.length;
        int[] waitingTime = new int[n];
        int[] turnaroundTime = new int[n];

        // Create a 2D array to store processes and their burst times
        Integer[][] processes = new Integer[n][2];
        for (int i = 0; i < n; i++) {
            processes[i][0] = i; // Process ID
            processes[i][1] = burstTimes[i]; // Burst Time
        }

        // Sort processes by burst time (SJF)
        Arrays.sort(processes, Comparator.comparingInt(a -> a[1]));

        // Calculate waiting time
        for (int i = 1; i < n; i++) {
            waitingTime[processes[i][0]] = waitingTime[processes[i - 1][0]] + processes[i - 1][1];
        }

        // Calculate turnaround time
        for (int i = 0; i < n; i++) {
            turnaroundTime[processes[i][0]] = waitingTime[processes[i][0]] + processes[i][1];
        }

        // Print results
        System.out.println("\nSJF Scheduling:");
        System.out.printf("%-10s %-10s %-10s %-10s\n", "Process", "Burst Time", "Waiting
Time", "Turnaround Time");
        int totalWaitingTime = 0;
        int totalTurnaroundTime = 0;
        for (int i = 0; i < n; i++) {
            totalWaitingTime += waitingTime[processes[i][0]];
            totalTurnaroundTime += turnaroundTime[processes[i][0]];
            System.out.printf("P%d\t\t%d\t\t%d\t\t%d\n", processes[i][0] + 1, processes[i][1],
waitingTime[processes[i][0]], turnaroundTime[processes[i][0]]);
        }

        System.out.printf("\nAverage Waiting Time: %.2f\n", (double) totalWaitingTime / n);
        System.out.printf("Average Turnaround Time: %.2f\n", (double) totalTurnaroundTime / n);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Process Scheduling Simulator");
        System.out.print("Enter the number of processes: ");
        int n = scanner.nextInt();
        int[] burstTimes = new int[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter burst time for process P" + (i + 1) + ": ");
            burstTimes[i] = scanner.nextInt();
        }

        System.out.print("Choose scheduling algorithm (1 for FCFS, 2 for SJF): ");
        int choice = scanner.nextInt();

        if (choice == 1) {
            calculateFCFS(burstTimes);
        } else if (choice == 2) {
            calculateSJF(burstTimes);
        } else {
            System.out.println("Invalid choice. Please choose either 1 or 2.");
        }

        scanner.close();
    }
}
```