Practical 7

AIM: **Write a program that implements FCFS scheduling algorithm.**

CODE:  
public class GFG {

static void findWaitingTime(int processes[], int n, int bt[], int wt[], int at[]) {

int service\_time[] = new int[n];

service\_time[0] = at[0];

wt[0] = 0;

for (int i = 1; i < n; i++) {

int wasted = 0;

service\_time[i] = service\_time[i - 1] + bt[i - 1];

wt[i] = service\_time[i] - at[i];

if (wt[i] < 0) {

wasted = Math.abs(wt[i]);

wt[i] = 0;

}

service\_time[i] = service\_time[i] + wasted;

}

}

static void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {

for (int i = 0; i < n; i++)

tat[i] = bt[i] + wt[i];

}

static void findavgTime(int processes[], int n, int bt[], int at[]) {

int wt[] = new int[n], tat[] = new int[n];

findWaitingTime(processes, n, bt, wt, at);

findTurnAroundTime(processes, n, bt, wt, tat);

System.out.println("Processes " + "Burst Time " + "Arrival Time " + "Waiting Time " + "Turn around time "

+ "Completion Time \n");

int total\_wt = 0, total\_tat = 0;

for (int i = 0; i < n; i++) {

total\_wt = total\_wt + wt[i];

total\_tat = total\_tat + tat[i];

int compl\_time = tat[i] + at[i];

System.out.println(

i + 1 + "\t\t" + bt[i] + "\t\t" + at[i] + "\t\t" + wt[i] + "\t\t" + tat[i] + "\t\t" + compl\_time);

}

System.out.println("Average waiting time=" + (float) total\_wt / (float) n);

System.out.println("\nAverage turn around time=" + (float) n);

}

public static void main(String args[]) {

int processes[] = { 1, 2, 3 };

int n = processes.length;

int burst\_time[] = { 5, 9, 6 };

int arrival\_time[] = { 0, 3, 6 };

findavgTime(processes, n, burst\_time, arrival\_time);

}

}

OUTPUT:

Processes Burst Time Arrival Time Waiting Time Turn around time Completion Time

1 5 0 0 5 5

2 9 3 2 11 14

3 6 6 8 14 20

Average waiting time=3.3333333

Average turn around time=3.0