

```

SJF.java
class SJF
{
    // Method to find the waiting time for all
    // processes
    static void findWaitingTime(Process proc[], int n, int wt[]){
        int rt[] = new int[n];

        // Copy the burst time into rt[]
        for (int i = 0; i < n; i++)
            rt[i] = proc[i].bt;

        int complete = 0, t = 0, minm = Integer.MAX_VALUE;
        int shortest = 0, finish_time;
        boolean check = false;

        // Process until all processes gets
        // completed
        while (complete != n) {

            // Find process with minimum
            // remaining time among the
            // processes that arrives till the
            // current time`
            for (int j = 0; j < n; j++)
            {
                if ((proc[j].art <= t) &&
                    (rt[j] < minm) && rt[j] > 0) {
                    minm = rt[j];
                    shortest = j;
                    check = true;
                }
            }

            if (check == false) {
                t++;
                continue;
            }

            // Reduce remaining time by one
            rt[shortest]--;

            // Update minimum
            minm = rt[shortest];
            if (minm == 0)
                minm = Integer.MAX_VALUE;

            // If a process gets completely
            // executed
            if (rt[shortest] == 0) {

                // Increment complete
                complete++;
                check = false;
            }
        }
    }
}

```

```

        // Find finish time of current
        // process
        finish_time = t + 1;

        // Calculate waiting time
        wt[shortest] = finish_time -
            proc[shortest].bt -
            proc[shortest].art;

        if (wt[shortest] < 0)
            wt[shortest] = 0;
    }
    // Increment time
    t++;
}

// Method to calculate turn around time
static void findTurnAroundTime(Process proc[], int n,
                                int wt[], int tat[])
{
    // calculating turn around time by adding
    // bt[i] + wt[i]
    for (int i = 0; i < n; i++)
        tat[i] = proc[i].bt + wt[i];
}

// Method to calculate average time
void findavgTime(Process proc[], int n)
{
    int wt[] = new int[n], tat[] = new int[n];
    int total_wt = 0, total_tat = 0;
//Function to find waiting time of all
// processes
    findWaitingTime(proc, n, wt);

// Function to find turn around time for
// all processes
    findTurnAroundTime(proc, n, wt, tat);

// Display processes along with all
// details
    System.out.println("Processes " +
        " Burst time " +
        " Waiting time " +
        " Turn around time");

// Calculate total waiting time and
// total turn around time
    for (int i = 0; i < n; i++) {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        System.out.println(" " + proc[i].pid + "\t\t\t"

```

```
        + proc[i].bt + "\t\t " + wt[i]
        + "\t\t" + tat[i]);
    }

    System.out.println("Average waiting time = " +
        (float)total_wt / (float)n);
    System.out.println("Average turn around time = " +
        (float)total_tat / (float)n);
}
}
```