SJF NON PREEMIPTIVES

import java.util.\*;

public class SJF {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of processes:");

int n = sc.nextInt();

int pid[] = new int[n];

int at[] = new int[n]; // Arrival time

int bt[] = new int[n]; // Burst time

int ct[] = new int[n]; // Completion time

int ta[] = new int[n]; // Turnaround time

int wt[] = new int[n]; // Waiting time

int f[] = new int[n]; // Flag to check if a process is completed

int st = 0, tot = 0;

float avgwt = 0, avgta = 0;

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

System.out.println("Enter process " + (i + 1) + " arrival time:");

at[i] = sc.nextInt();

System.out.println("Enter process " + (i + 1) + " burst time:");

bt[i] = sc.nextInt();

pid[i] = i + 1;

f[i] = 0;

}

while (true) {

int c = n, min = 999;

if (tot == n) // If the total number of processes is completed, the loop terminates

break;

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

/\*

\* If i'th process arrival time <= system time and its flag=0 and burst<min

\* That process will be executed first

\*/

if ((at[i] <= st) && (f[i] == 0) && (bt[i] < min)) {

min = bt[i];

c = i;

}

}

/\* If c==n means c value cannot be updated because no process

arrival time < system time so we increase the system time \*/

if (c == n)

st++;

else {

ct[c] = st + bt[c];

st += bt[c];

ta[c] = ct[c] - at[c];

wt[c] = ta[c] - bt[c];

f[c] = 1;

tot++;

}

}

System.out.println("\npid arrival burst complete turnaround waiting");

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

avgwt += wt[i];

avgta += ta[i];

System.out.println(pid[i] + "\t" + at[i] + "\t" + bt[i] + "\t" + ct[i] + "\t" + ta[i] + "\t" + wt[i]);

}

System.out.println("\nAverage turnaround time: " + (float) (avgta / n));

System.out.println("Average waiting time: " + (float) (avgwt / n));

sc.close();

}

}