//sjf.c

#include <stdio.h>

int main() {

int n;

// Taking input for number of processes

printf("Enter the number of processes: ");

scanf("%d", &n);

int pid[n], bt[n], at[n], wt[n], tat[n], completed[n];

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

// Accept Process IDs using a for loop

printf("Enter Process IDs: \n");

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

printf("Enter Process ID for Process %d: ", i + 1);

scanf("%d", &pid[i]);

}

// Accept Burst Times using a separate for loop

printf("Enter Burst Times: \n");

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

printf("Enter Burst Time for Process %d: ", i + 1);

scanf("%d", &bt[i]);

}

// Accept Arrival Times using a separate for loop

printf("Enter Arrival Times: \n");

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

printf("Enter Arrival Time for Process %d: ", i + 1);

scanf("%d", &at[i]);

completed[i] = 0; // Initially, no process is completed

}

// Calculate waiting time and turnaround time for each process

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

int idx = -1;

int min\_bt = 9999;

// Find the process with the smallest burst time that has arrived and not yet completed

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

if (!completed[i] && at[i] <= current\_time && bt[i] < min\_bt) {

min\_bt = bt[i];

idx = i;

}

}

// If no process can be executed (i.e., all are waiting), move the current time forward

if (idx == -1) {

current\_time++;

count--;

continue;

}

// Calculate the waiting time and turnaround time for the selected process

wt[idx] = current\_time - at[idx];

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

// Update total waiting time and total turnaround time

total\_wt += wt[idx];

total\_tat += tat[idx];

// Update the current time

current\_time += bt[idx];

// Mark the process as completed

completed[idx] = 1;

}

// Output the results

printf("\nProcess ID\tBurst Time\tArrival Time\tWaiting Time\tTurnaround Time\n");

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

printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", pid[i], bt[i], at[i], wt[i], tat[i]);

}

// Calculating and printing average waiting time and average turnaround time

printf("\nAverage Waiting Time: %.2f\n", (float)total\_wt / n);

printf("Average Turnaround Time: %.2f\n", (float)total\_tat / n);

return 0;

}

Enter the number of processes: 5

Enter Process IDs:

Enter Process ID for Process 1: 1

Enter Process ID for Process 2: 2

Enter Process ID for Process 3: 3

Enter Process ID for Process 4: 4

Enter Process ID for Process 5: 5

Enter Burst Times:

Enter Burst Time for Process 1: 5

Enter Burst Time for Process 2: 6

Enter Burst Time for Process 3: 7

Enter Burst Time for Process 4: 8

Enter Burst Time for Process 5: 9

Enter Arrival Times:

Enter Arrival Time for Process 1: 0

Enter Arrival Time for Process 2: 1

Enter Arrival Time for Process 3: 2

Enter Arrival Time for Process 4: 3

Enter Arrival Time for Process 5: 4

Process ID Burst Time Arrival Time Waiting Time Turnaround Time

1 5 0 0 5

2 6 1 4 10

3 7 2 9 16

4 8 3 15 23

5 9 4 22 31

Average Waiting Time: 10.00

Average Turnaround Time: 17.00