#include <stdio.h>

struct Process {

int pid;

int burst\_time;

int waiting\_time;

int turn\_around\_time;

};

void sortProcessesByBurstTime(struct Process proc[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (proc[j].burst\_time > proc[j + 1].burst\_time) {

struct Process temp = proc[j];

proc[j] = proc[j + 1];

proc[j + 1] = temp;

}

}

}

}

void calculateTimes(struct Process proc[], int n) {

proc[0].waiting\_time = 0;

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

proc[i].waiting\_time = proc[i - 1].waiting\_time + proc[i - 1].burst\_time;

}

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

proc[i].turn\_around\_time = proc[i].waiting\_time + proc[i].burst\_time;

}

}

void printProcessInfo(struct Process proc[], int n) {

printf("PID\tBurst\tWait\tTurnaround\n");

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

printf("%d\t%d\t%d\t%d\n", proc[i].pid, proc[i].burst\_time, proc[i].waiting\_time, proc[i].turn\_around\_time);

}

}

int main() {

int n;

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

scanf("%d", &n);

struct Process proc[n];

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

proc[i].pid = i + 1;

printf("Enter burst time for process %d: ", i + 1);

scanf("%d", &proc[i].burst\_time);

}

sortProcessesByBurstTime(proc, n);

calculateTimes(proc, n);

printProcessInfo(proc, n);

float total\_waiting\_time = 0, total\_turnaround\_time = 0;

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

total\_waiting\_time += proc[i].waiting\_time;

total\_turnaround\_time += proc[i].turn\_around\_time;

}

printf("Average waiting time = %.2f\n", total\_waiting\_time / n);

printf("Average turnaround time = %.2f\n", total\_turnaround\_time / n);

return 0;

}

