#include <stdio.h>

struct Process {

int pid;

int burst\_time;

int remaining\_time;

int waiting\_time;

int turn\_around\_time;

};

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

int current\_time = 0;

int completed = 0;

while (completed < n) {

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

if (proc[i].remaining\_time > 0) {

if (proc[i].remaining\_time > quantum) {

current\_time += quantum;

proc[i].remaining\_time -= quantum;

} else {

current\_time += proc[i].remaining\_time;

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

proc[i].remaining\_time = 0;

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

completed++;

}

}

}

}

}

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, quantum;

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

scanf("%d", &n);

printf("Enter the time quantum: ");

scanf("%d", &quantum);

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);

proc[i].remaining\_time = proc[i].burst\_time;

proc[i].waiting\_time = 0;

proc[i].turn\_around\_time = 0;

}

calculateTimes(proc, n, quantum);

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;

}

