#include <stdio.h>

struct Process {

int id; // Process ID

int burst\_time; // Burst time

};

void calculateTimes(struct Process processes[], int n, int waiting\_time[], int turnaround\_time[]) {

waiting\_time[0] = 0;

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

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

}

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

turnaround\_time[i] = waiting\_time[i] + processes[i].burst\_time;

}

}

void printSchedule(struct Process processes[], int n, int waiting\_time[], int turnaround\_time[]) {

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

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

printf("%d\t\t%d\t\t%d\t\t%d\n", processes[i].id, processes[i].burst\_time, waiting\_time[i], turnaround\_time[i]);

}

}

int main() {

int n;

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

scanf("%d", &n);

struct Process processes[n];

int waiting\_time[n], turnaround\_time[n];

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

processes[i].id = i + 1;

printf("Enter burst time for Process %d: ", processes[i].id);

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

}

calculateTimes(processes, n, waiting\_time, turnaround\_time);

printSchedule(processes, n, waiting\_time, turnaround\_time);

return 0;

}

#include <stdio.h>

struct Process {

int id; // Process ID

int burst\_time; // Burst time

};

void calculateTimes(struct Process processes[], int n, int waiting\_time[], int turnaround\_time[]) {

waiting\_time[0] = 0;

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

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

}

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

turnaround\_time[i] = waiting\_time[i] + processes[i].burst\_time;

}

}

void printSchedule(struct Process processes[], int n, int waiting\_time[], int turnaround\_time[]) {

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

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

printf("%d\t\t%d\t\t%d\t\t%d\n", processes[i].id, processes[i].burst\_time, waiting\_time[i], turnaround\_time[i]);

}

}

int main() {

int n;

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

scanf("%d", &n);

struct Process processes[n];

int waiting\_time[n], turnaround\_time[n];

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

processes[i].id = i + 1;

printf("Enter burst time for Process %d: ", processes[i].id);

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

}

calculateTimes(processes, n, waiting\_time, turnaround\_time);

printSchedule(processes, n, waiting\_time, turnaround\_time);

return 0;

}#include <stdio.h>

struct Process {

int id; // Process ID

int burst\_time; // Burst time

};

void calculateTimes(struct Process processes[], int n, int waiting\_time[], int turnaround\_time[]) {

waiting\_time[0] = 0;

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

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

}

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

turnaround\_time[i] = waiting\_time[i] + processes[i].burst\_time;

}

}

void printSchedule(struct Process processes[], int n, int waiting\_time[], int turnaround\_time[]) {

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

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

printf("%d\t\t%d\t\t%d\t\t%d\n", processes[i].id, processes[i].burst\_time, waiting\_time[i], turnaround\_time[i]);

}

}

int main() {

int n;

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

scanf("%d", &n);

struct Process processes[n];

int waiting\_time[n], turnaround\_time[n];

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

processes[i].id = i + 1;

printf("Enter burst time for Process %d: ", processes[i].id);

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

}

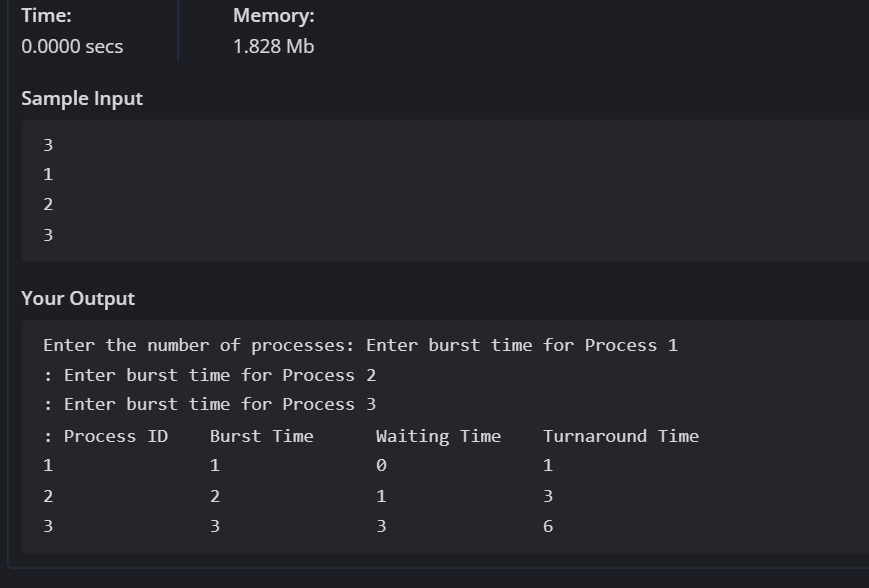
calculateTimes(processes, n, waiting\_time, turnaround\_time);

printSchedule(processes, n, waiting\_time, turnaround\_time);

return 0;

}

**Input and Output:**

****