**Name : Pranav Jawale**

**Roll No. TIA28**

**CODE >>**

#include<stdio.h>

#include<stdlib.h>

struct Process {

int id;

int arrival\_time;

int service\_time;

int waiting\_time;

int turnaround\_time;

int remaining\_time;

};

void shortestJobFirst(struct Process processes[], int n) {

int currentTime = 0;

int completed = 0;

int shortest;

while (completed < n) {

shortest = -1;

int minServiceTime = INT\_MAX;

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

if (processes[i].arrival\_time <= currentTime && processes[i].remaining\_time > 0 && processes[i].service\_time < minServiceTime) {

shortest = i;

minServiceTime = processes[i].service\_time;

}

}

if (shortest == -1) {

currentTime++;

} else {

processes[shortest].remaining\_time--;

currentTime++;

if (processes[shortest].remaining\_time == 0) {

completed++;

processes[shortest].turnaround\_time = currentTime - processes[shortest].arrival\_time;

processes[shortest].waiting\_time = processes[shortest].turnaround\_time - processes[shortest].service\_time;

}

}

}

}

int main() {

int n;

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

scanf("%d", &n);

struct Process processes[n];

// Input arrival time and service time for each process

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

processes[i].id = i + 1;

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

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

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

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

processes[i].remaining\_time = processes[i].service\_time;

}

shortestJobFirst(processes, n);

printf("Process\tArrival Time\tService Time\tWaiting Time\tTurnaround Time\n");

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

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

}

// Calculate average waiting time and average turnaround time

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

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

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

total\_turnaround\_time += processes[i].turnaround\_time;

}

float avg\_waiting\_time = total\_waiting\_time / n;

float avg\_turnaround\_time = total\_turnaround\_time / n;

printf("Average Waiting Time: %.2f\n", avg\_waiting\_time);

printf("Average Turnaround Time: %.2f\n", avg\_turnaround\_time);

return 0;

}

**Output >>**

Enter the number of processes: 5

Enter arrival time for process 1: 0

Enter service time for process 1: 3

Enter arrival time for process 2: 2

Enter service time for process 2: 6

Enter arrival time for process 3: 4

Enter service time for process 3: 4

Enter arrival time for process 4: 6

Enter service time for process 4: 5

Enter arrival time for process 5: 8

Enter service time for process 5: 2

Process Arrival Time Service Time Waiting Time Turnaround Time

1 0 3 0 3

2 2 6 12 18

3 4 4 0 4

4 6 5 4 9

5 8 2 0 2

Average Waiting Time: 3.20

Average Turnaround Time: 7.20

PS P:\VsCode>