

**PROGRAM:**

#include <stdio.h>

struct Process {

    int processID;

    int burstTime;

    int priority;

    int waitingTime;

    int turnAroundTime;

};

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

    struct Process temp;

    for (int i = 0; i < n - 1; i++) {

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

            if (processes[i].priority > processes[j].priority) {

                temp = processes[i];

                processes[i] = processes[j];

                processes[j] = temp;

            }

        }

    }

}

int main() {

    int n;

    float totalWaitingTime = 0, totalTurnAroundTime = 0;

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

    scanf("%d", &n);

    struct Process processes[n];

    printf("Enter the burst time and priority of the processes:\n");

    for (int i = 0; i < n; i++) {

        processes[i].processID = i + 1;

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

        scanf("%d", &processes[i].burstTime);

        printf("Process %d - Priority: ", i + 1);

        scanf("%d", &processes[i].priority);

        processes[i].waitingTime = 0;

        processes[i].turnAroundTime = 0;

    }

    sortProcesses(processes, n);

    processes[0].waitingTime = 0; // The waiting time of the first process is always 0

    for (int i = 1; i < n; i++) {

        processes[i].waitingTime = processes[i - 1].burstTime + processes[i - 1].waitingTime;

    }

    for (int i = 0; i < n; i++) {

        processes[i].turnAroundTime = processes[i].burstTime + processes[i].waitingTime;

    }

    printf("\nProcess Burst Time Priority Waiting Time Turn Around Time\n");

    for (int i = 0; i < n; i++) {

        printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", processes[i].processID, processes[i].burstTime,

               processes[i].priority, processes[i].waitingTime, processes[i].turnAroundTime);

        totalWaitingTime += processes[i].waitingTime;

        totalTurnAroundTime += processes[i].turnAroundTime;

    }

    printf("\nAverage waiting time is: %.2f\n", totalWaitingTime / n);

    printf("Average Turn Around Time is: %.2f\n", totalTurnAroundTime / n);

    return 0;

}

**OUTPUT:**

Enter the number of processes: 2

Enter the burst time and priority of the processes:

Process 1 - Burst Time: 3

Process 1 - Priority: 2

Process 2 - Burst Time: 4

Process 2 - Priority: 1

Process Burst Time Priority Waiting Time Turn Around Time

2 4 1 0 4

1 3 2 4 7

Average waiting time is: 2.00

Average Turn Around Time is: 5.50