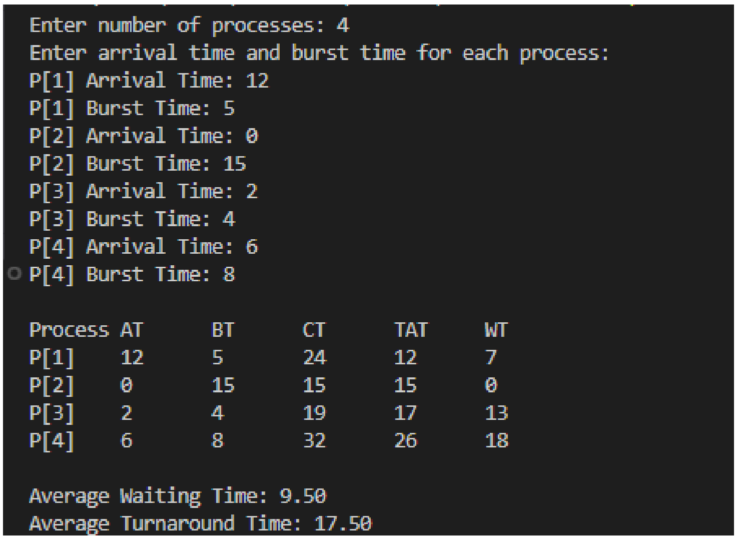
OS LAB SJF

NON PREEMPTIVE

```
#include <stdio.h>

#define MAX_PROCESSES 10

int main() {
    int n, at[MAX_PROCESSES], bt[MAX_PROCESSES], wt[MAX_PROCESSES],
tat[MAX_PROCESSES], ct[MAX_PROCESSES];
    int completed[MAX_PROCESSES] = {0}, time = 0, smallest;
    float sum_wt = 0, sum_tat = 0;

    printf("Enter number of processes: ");
    scanf("%d", &n);

    printf("Enter arrival time and burst time for each process:\n");
    for (int i = 0; i < n; i++) {
        printf("P[%d] Arrival Time: ", i + 1);
        scanf("%d", &at[i]);
        printf("P[%d] Burst Time: ", i + 1);
        scanf("%d", &bt[i]);
    }

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

        for (int j = 0; j < n; j++) {
            if (at[j] <= time && !completed[j]) {
                if (smallest == -1 || bt[j] < bt[smallest]) {
                    smallest = j;
                }
            }
        }

        if (smallest == -1) {
            time++;
            continue;
        }

        time += bt[smallest];
        ct[smallest] = time;
        tat[smallest] = ct[smallest] - at[smallest];
        wt[smallest] = tat[smallest] - bt[smallest];

        sum_wt += wt[smallest];
        sum_tat += tat[smallest];
        completed[smallest] = 1;
    }
    printf("\nProcess\tAT\tBT\tCT\tTAT\tWT\n");
    for (int i = 0; i < n; i++) {
        printf("P[%d]\t%d\t%d\t%d\t%d\t%d\n", i + 1, at[i], bt[i], ct[i], tat[i], wt[i]);
    }

    printf("\nAverage Waiting Time: %.2f\n", sum_wt / n);
    printf("Average Turnaround Time: %.2f\n", sum_tat / n);

    return 0;
}
```

Output:

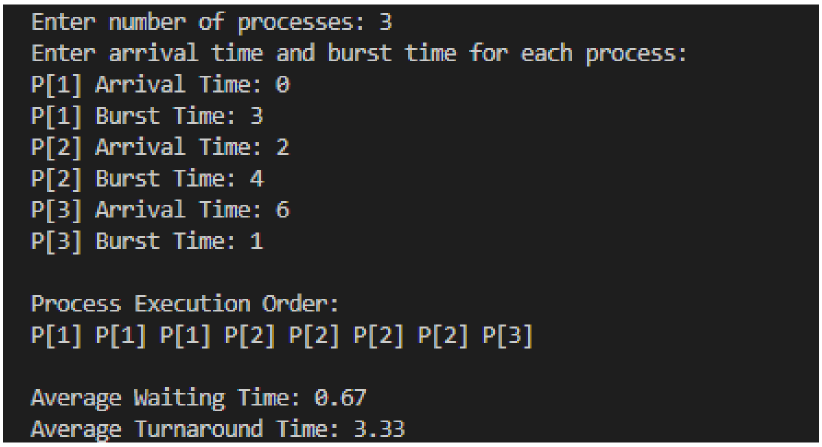


```
Enter number of processes: 4
Enter arrival time and burst time for each process:
P[1] Arrival Time: 12
P[1] Burst Time: 5
P[2] Arrival Time: 0
P[2] Burst Time: 15
P[3] Arrival Time: 2
P[3] Burst Time: 4
P[4] Arrival Time: 6
P[4] Burst Time: 8

Process AT      BT      CT      TAT     WT
P[1]    12      5       24      12      7
P[2]    0       15      15      15      0
P[3]    2       4       19      17      13
P[4]    6       8       32      26      18

Average Waiting Time: 9.50
Average Turnaround Time: 17.50
```

PREEMPTIVE

```
#include <stdio.h>

#define MAX_PROCESSES 10

int main() {
    int n, at[MAX_PROCESSES], bt[MAX_PROCESSES], rt[MAX_PROCESSES];
    int complete = 0, time = 0, smallest = -1;
    int min_rt, finish_time, sum_wait = 0, sum_turnaround = 0;

    printf("Enter number of processes: ");
    scanf("%d", &n);

    printf("Enter arrival time and burst time for each process:\n");
    for (int i = 0; i < n; i++) {
        printf("P[%d] Arrival Time: ", i + 1);
        scanf("%d", &at[i]);
        printf("P[%d] Burst Time: ", i + 1);
        scanf("%d", &bt[i]);
        rt[i] = bt[i];
    }
    printf("\nProcess Execution Order:\n");

    while (complete < n) {
        smallest = -1;
        min_rt = 9999;

        for (int i = 0; i < n; i++) {
            if (at[i] <= time && rt[i] > 0 && rt[i] < min_rt) {
                min_rt = rt[i];
                smallest = i;
            }
        }

        if (smallest == -1) {
            time++;
            continue;
        }

        rt[smallest]--;
        printf("P[%d] ", smallest + 1);

        if (rt[smallest] == 0) {
            complete++;
            finish_time = time + 1;
            sum_wait += finish_time - at[smallest] - bt[smallest];
            sum_turnaround += finish_time - at[smallest];
        }

        time++;
    }

    printf("\n\nAverage Waiting Time: %.2f\n", (float)sum_wait / n);
    printf("Average Turnaround Time: %.2f\n", (float)sum_turnaround / n);

    return 0;
}
```

Output:



```
Enter number of processes: 3
Enter arrival time and burst time for each process:
P[1] Arrival Time: 0
P[1] Burst Time: 3
P[2] Arrival Time: 2
P[2] Burst Time: 4
P[3] Arrival Time: 6
P[3] Burst Time: 1

Process Execution Order:
P[1] P[1] P[1] P[2] P[2] P[2] P[2] P[3]

Average Waiting Time: 0.67
Average Turnaround Time: 3.33
```