# **LAB-02**

**Exercise:**

1) Implement the First Come First Serve code and paste the output below.

**CODE:**

#include <stdio.h>

int main()

{

    int at[20], bt[20], wt[20], tat[20], ct[20], i, n;

    float wtavg = 0, tatavg = 0;

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

    scanf("%d", &n);

    for (i = 0; i < n; i++)

    {

        printf("Enter Arrival Time and Burst Time for Process P%d: ", i + 1);

        scanf("%d%d", &at[i], &bt[i]);

    }

    ct[0] = at[0] + bt[0];

    tat[0] = ct[0] - at[0];

    wt[0] = tat[0] - bt[0];

    for (i = 1; i < n; i++)

    {

        if (at[i] > ct[i - 1])

        {

            ct[i] = at[i] + bt[i];

        }

        else

        {

            ct[i] = ct[i - 1] + bt[i];

        }

        tat[i] = ct[i] - at[i];

        wt[i] = tat[i] - bt[i];

    }

    for (i = 0; i < n; i++)

    {

        wtavg += wt[i];

        tatavg += tat[i];

    }

    wtavg /= n;

    tatavg /= n;

    printf("\nProcess\tArrival Time\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");

    for (i = 0; i < n; i++)

    {

        printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, at[i], bt[i], ct[i], tat[i], wt[i]);

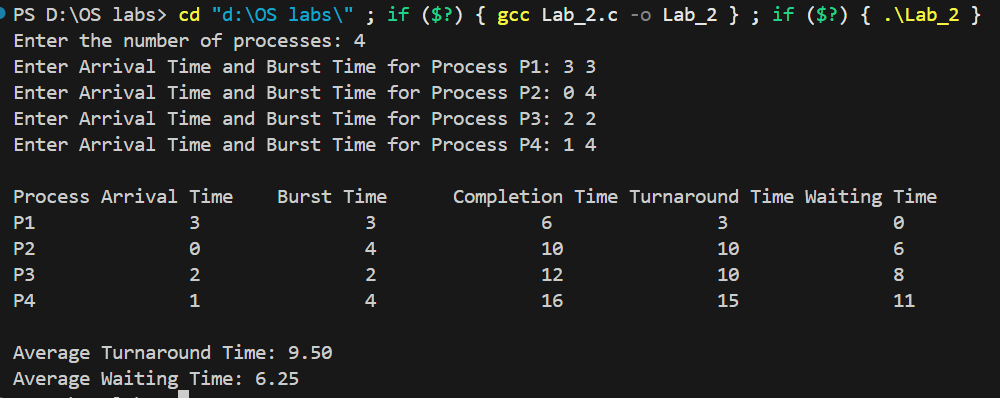
    }

    printf("\nAverage Turnaround Time: %.2f\n", tatavg);

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

    return 0;

}

**OUTPUT:**

2) Implement the Shortest Job First code and paste the output below.

**CODE:**

#include <stdio.h>

int main()

{

    int p[20], at[20], bt[20], wt[20], tat[20], ct[20], i, k, n, temp;

    float wtavg = 0, tatavg = 0;

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

    scanf("%d", &n);

    for (i = 0; i < n; i++)

    {

        p[i] = i + 1;

        printf("Enter Arrival Time for Process P%d: ", i + 1);

        scanf("%d", &at[i]);

        printf("Enter Burst Time for Process P%d: ", i + 1);

        scanf("%d", &bt[i]);

    }

    for (i = 0; i < n; i++)

    {

        for (k = i + 1; k < n; k++)

        {

            if (at[i] > at[k] || (at[i] == at[k] && bt[i] > bt[k]))

            {

                temp = at[i];

                at[i] = at[k];

                at[k] = temp;

                temp = bt[i];

                bt[i] = bt[k];

                bt[k] = temp;

                temp = p[i];

                p[i] = p[k];

                p[k] = temp;

            }

        }

    }

    ct[0] = at[0] + bt[0];

    tat[0] = ct[0] - at[0];

    wt[0] = tat[0] - bt[0];

    wtavg = wt[0];

    tatavg = tat[0];

    for (i = 1; i < n; i++)

    {

        if (ct[i - 1] < at[i])

        {

            ct[i] = at[i] + bt[i];

        }

        else

        {

            ct[i] = ct[i - 1] + bt[i];

        }

        tat[i] = ct[i] - at[i];

        wt[i] = tat[i] - bt[i];

        wtavg += wt[i];

        tatavg += tat[i];

    }

    wtavg /= n;

    tatavg /= n;

    printf("\nProcess\tArrival Time\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");

    for (i = 0; i < n; i++)

    {

        printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", p[i], at[i], bt[i], ct[i], tat[i], wt[i]);

    }

    printf("\nAverage Turnaround Time: %.2f\n", tatavg);

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

    return 0;

}

**OUTPUT:**

