Course: Operating System
Course code: CT-353

Lab 02

QUESTION 01: Implement the First Come First Serve code and paste the output below.

ANSWER:

CODE:

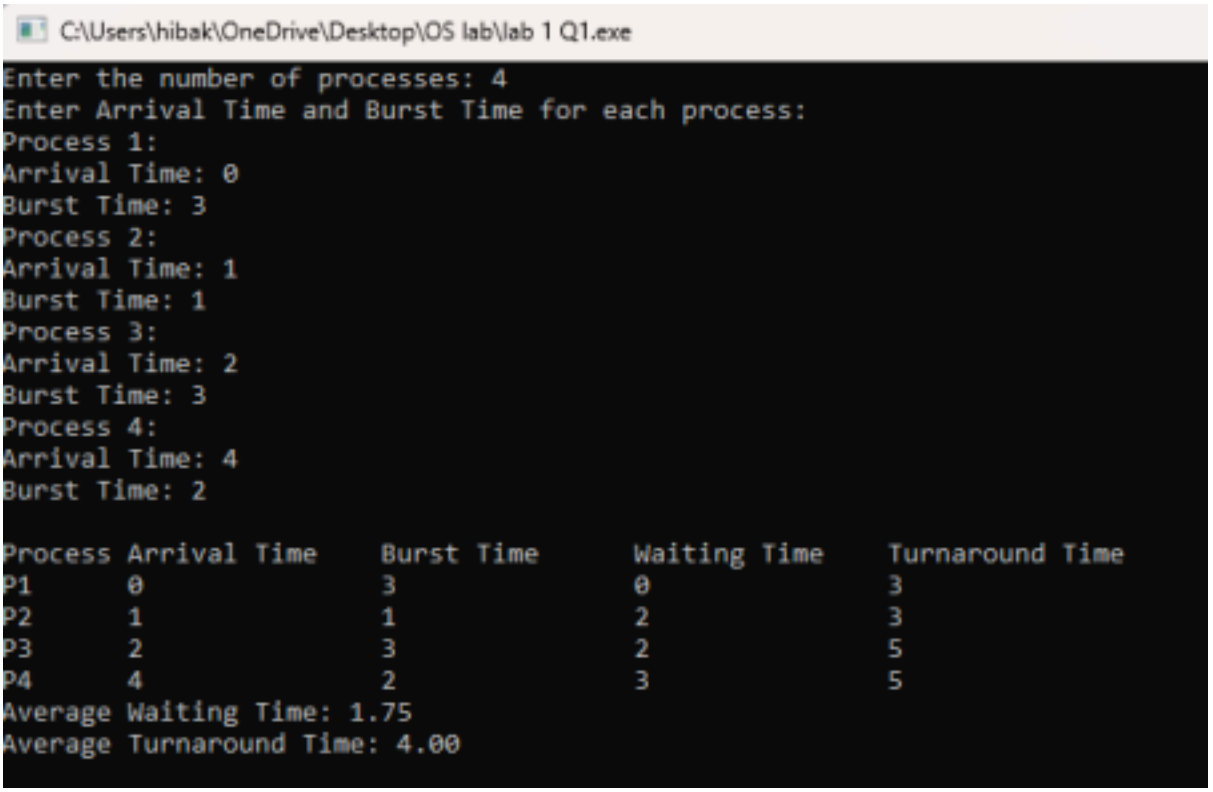
```
#include <stdio.h>

int main() {
    int n, i, j;
    int at[20], bt[20], wt[20], tat[20], ct[20]; // AT = Arrival Time, CT = Completion Time
    float wtavg = 0, tatavg = 0;

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

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

    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (at[i] > at[j]) {

                int temp = at[i];
                at[i] = at[j];
                at[j] = temp;

                temp = bt[i];
                bt[i] = bt[j];
                bt[j] = temp;
            }
        }
    }

    // Calculate Completion Time, Waiting Time, and Turnaround Time.
    ct[0] = at[0] + bt[0];
    tat[0] = ct[0] - at[0];
    wt[0] = tat[0] - bt[0];

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

Name: Hiba Kazmi
Roll no: DT-22025

Course: Operating System
Course code: CT-353

```
        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];
    }

    for (i = 0; i < n; i++) {
        wtavg += wt[i];
        tatavg += tat[i];
    }

    // Display results.
    printf("\nProcess\tArrival Time\tBurst Time\tWaiting Time\tTurnaround
    Time\n"); for (i = 0; i < n; i++) {
        printf("P%d\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, at[i], bt[i], wt[i], tat[i]);
    }

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

    return 0;
}
```

OUTPUT:



Name: Hiba Kazmi
Roll no: DT-22025

Course: Operating System
Course code: CT-353

QUESTION 02: Implement the Shortest Job First code and paste the output below.

ANSWER:

CODE:

```
#include <stdio.h>

int main() {
    int p[20], at[20], bt[20], wt[20], tat[20], ct[20], i, j, n, temp;
    float wtavg = 0, tatavg = 0;

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

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

    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if ((at[i] > at[j]) || (at[i] == at[j] && bt[i] > bt[j])) {
                temp = at[i];
                at[i] = at[j];
                at[j] = temp;

                temp = bt[i];
                bt[i] = bt[j];
                bt[j] = temp;

                temp = p[i];
                p[i] = p[j];
                p[j] = temp;
            }
        }
    }

    ct[0] = at[0] + bt[0];
    tat[0] = ct[0] - at[0];
    wt[0] = tat[0] - bt[0];
```

Name: Hiba Kazmi
Roll no: DT-22025

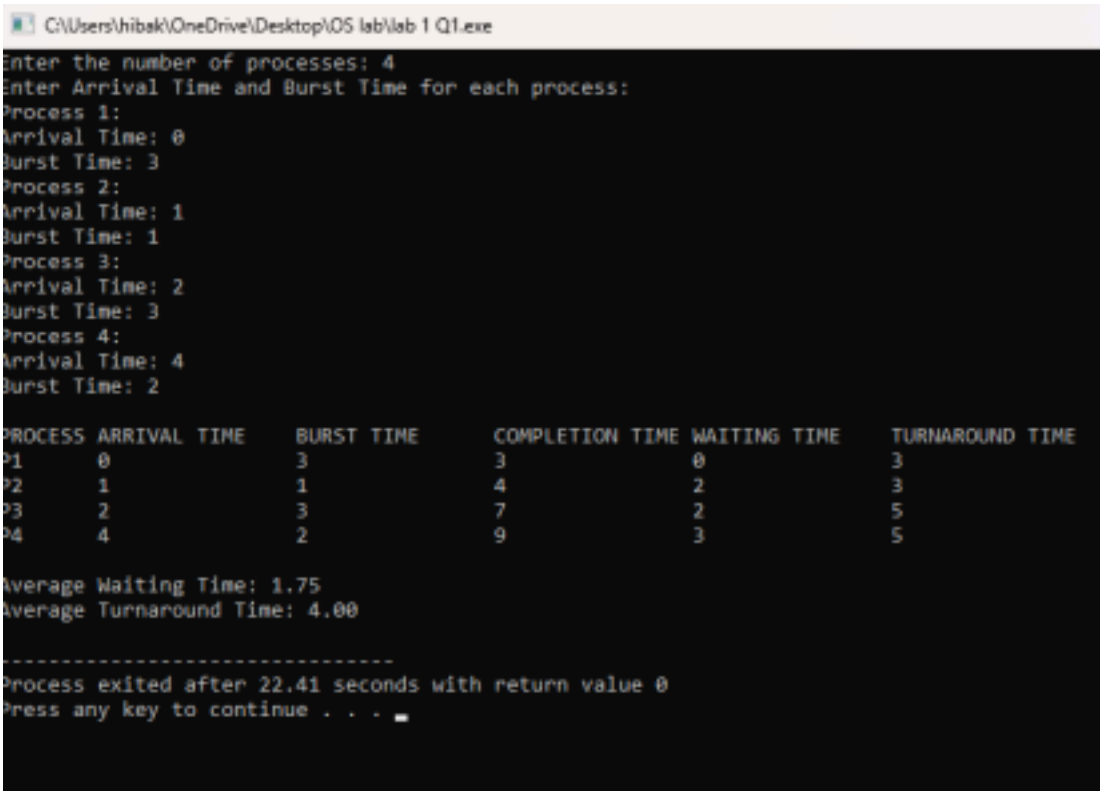
Course: Operating System
Course code: CT-353

```
    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];
    }

    for (i = 0; i < n; i++) {
        wtavg += wt[i];
        tatavg += tat[i];
    }

    // Display results
    printf("\nPROCESS\tARRIVAL TIME\tBURST TIME\tCOMPLETION TIME\tWAITING
TIME\tTURNAROUND TIME\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", p[i], at[i], bt[i], ct[i], wt[i],
    tat[i]); }

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

    return 0;
}
```

OUTPUT:

Name: Hiba Kazmi
Roll no: DT-22025

Course: Operating System
Course code: CT-353



Name: Hiba Kazmi
Roll no: DT-22025