Operating Systems – Lab#2 SOLUTIONS

```c
/* FIRST COME FIRST SERVE ALGORITHM */
#include<stdio.h>

int main()
{
    int bt[20], wt[20], tat[20], i, n;
    float wtavg, tatavg;

    printf("\nEnter the number of processes -- ");
    scanf_s("%d", &n);
    for (i = 0; i < n; i++)
    {
        printf("\nEnter Burst Time for Process %d -- ", i);
        scanf_s("%d", &bt[i]);
    }
    wt[0] = wtavg = 0;
    tat[0] = tatavg = bt[0];
    for (i = 1; i < n; i++)
    {
        wt[i] = wt[i - 1] + bt[i - 1];
        tat[i] = tat[i - 1] + bt[i];
        wtavg = wtavg + wt[i];
        tatavg = tatavg + tat[i];
    }
    printf("\t PROCESS \tBURST TIME \t WAITING TIME\t TURNAROUND TIME\n");
    for (i = 0; i < n; i++)
        printf("\n\t P%d \t\t %d \t\t %d \t\t %d", i, bt[i], wt[i], tat[i]);
    printf("\nAverage Waiting Time -- %f", wtavg / n);
    printf("\nAverage Turnaround Time -- %f", tatavg / n);

}
```

Microsoft Visual Studio Debug Console

```
Enter the number of processes -- 4

Enter Burst Time for Process 0 -- 25

Enter Burst Time for Process 1 -- 45

Enter Burst Time for Process 2 -- 82

Enter Burst Time for Process 3 -- 34
        PROCESS         BURST TIME      WAITING TIME    TURNAROUND TIME

        P0              25              0               25
        P1              45              25              70
        P2              82              70              152
        P3              34              152             186
Average Waiting Time -- 61.750000
Average Turnaround Time -- 108.250000
```

Operating Systems – Lab#2 SOLUTIONS

```c
/* Shortest Job First */
#include<stdio.h>
int main()
{
    int p[20], bt[20], wt[20], tat[20], i, k, n, temp;
    float wtavg, tatavg;

    printf("\nEnter the number of processes -- ");
    scanf_s("%d", &n);
    for (i = 0; i < n; i++)
    {
        p[i] = i;
        printf("Enter Burst Time for Process %d -- ", i);
        scanf_s("%d", &bt[i]);
    }
    for (i = 0; i < n; i++)
        for (k = i + 1; k < n; k++)
            if (bt[i] > bt[k])
            {
                temp = bt[i];
                bt[i] = bt[k];
                bt[k] = temp;
                temp = p[i];
                p[i] = p[k];
                p[k] = temp;
            }
    wt[0] = wtavg = 0;
    tat[0] = tatavg = bt[0];
    for (i = 1; i < n; i++)
    {
        wt[i] = wt[i - 1] + bt[i - 1];
        tat[i] = tat[i - 1] + bt[i];
        wtavg = wtavg + wt[i];
        tatavg = tatavg + tat[i];
    }

    printf("\n\t PROCESS \tBURST TIME \t WAITING TIME\t TURNAROUND TIME\n");
    for (i = 0; i < n; i++)
        printf("\n\t P%d \t\t %d \t\t %d \t\t %d", p[i], bt[i], wt[i], tat[i]);
    printf("\nAverage Waiting Time -- %f", wtavg / n);
    printf("\nAverage Turnaround Time -- %f", tatavg / n);
}
```

Microsoft Visual Studio Debug Console

```
Enter the number of processes -- 4
Enter Burst Time for Process 0 -- 25
Enter Burst Time for Process 1 -- 45
Enter Burst Time for Process 2 -- 82
Enter Burst Time for Process 3 -- 34

        PROCESS         BURST TIME      WAITING TIME    TURNAROUND TIME

        P0              25              0               25
        P3              34              25              59
        P1              45              59              104
        P2              82              104             186
Average Waiting Time -- 47.000000
Average Turnaround Time -- 93.500000
```