Lab Assignment 1
Chaudhary Hamdan
1905387
Date: 07-01-2022

Question:
Scheduling
Algorithm(RR,SJF(preemptive&nonpreemptive),FCFS)

1. FCFS:
Code:

```c
// Author: Chaudhary Hamdan, 1905387

#include<stdio.h>


void findWaitingTime(int processes[], int n, int bt[], int wt[])
{
    wt[0] = 0;

    for (int i = 1; i < n ; i++ )
        wt[i] = bt[i - 1] + wt[i - 1] - processes[i];
}

void findTurnAroundTime( int processes[], int n, int bt[], int wt[], int tat[])
{
    for (int i = 0; i < n ; i++)
        tat[i] = bt[i] + wt[i];
}

void findavgTime( int processes[], int n, int bt[])
{
    int wt[n], tat[n], total_wt = 0, total_tat = 0;

    findWaitingTime(processes, n, bt, wt);
```

```c
        findTurnAroundTime(processes, n, bt, wt, tat);


        for (int i = 0; i < n; i++)
        {
                total_wt = total_wt + wt[i];
                total_tat = total_tat + tat[i];
        }
        int s = total_wt * 1.0 / n;
        int t = total_tat * 1.0 / n;
        printf("Average waiting time: %d", s);
        printf("\nAverage turn around time: %d ", t);
}

int main()
{
#ifndef ONLINE_JUDGE
        freopen("C:\\Users\\KIIT\\input", "r", stdin);
        freopen("C:\\Users\\KIIT\\output", "w", stdout);
#endif


        int n;
        scanf("%d", &n);

        int processes[n], burst_time[n];

        for (int i = 0; i < n ; i++) {
                scanf("%d", &processes[i]);
                scanf("%d", &burst_time[i]);
        }

        findavgTime(processes, n, burst_time);
        return 0;
}
```
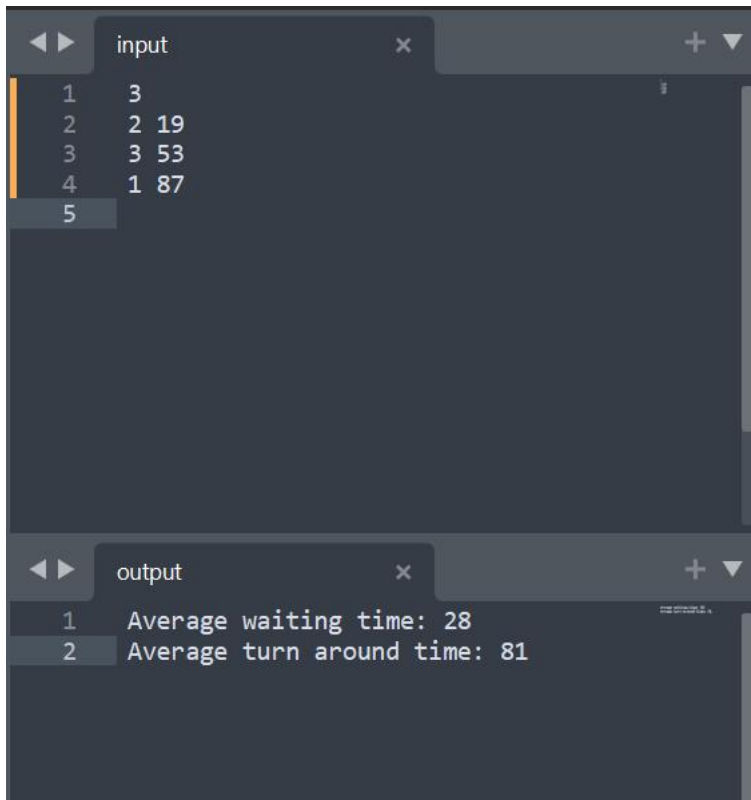
Output:

```
input                           ×        + ▼
1    3
2    2 19
3    3 53
4    1 87
5
```

```
output                          ×        + ▼
1    Average waiting time: 28
2    Average turn around time: 81
```

2. SJF (Preemptive)
Code:

```
// Author: Chaudhary Hamdan, 1905387

#include <stdio.h>


int main()
{
#ifndef ONLINE_JUDGE
    freopen("C:\\Users\\KIIT\\input", "r", stdin);
    freopen("C:\\Users\\KIIT\\output", "w", stdout);
#endif



    int arrival_time[10], burst_time[10], temp[10];
    int i, smallest, count = 0, time, limit;
    double wait_time = 0, turnaround_time = 0, end;
    float average_waiting_time, average_turnaround_time;
    scanf("%d", &limit);
    for (i = 0; i < limit; i++)
    {
        scanf("%d", &arrival_time[i]);
        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }
    burst_time[9] = 9999;
    for (time = 0; count < limit; time++)
    {
        smallest = 9;
        for (i = 0; i < limit; i++)
        {
            if (arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0)
            {
                smallest = i;
            }
        }
```

```c
            burst_time[smallest]--;
            if (burst_time[smallest] == 0)
            {
                    count++;
                    end = time + 1;
                    wait_time = wait_time + end - arrival_time[smallest] -
temp[smallest];
                    turnaround_time = turnaround_time + end -
arrival_time[smallest];
            }
    }
    average_waiting_time = wait_time / limit;
    average_turnaround_time = turnaround_time / limit;
    printf("Average Waiting Time:%lf\n", average_waiting_time);
    printf("Average Turnaround Time:%lf\n", average_turnaround_time);
    return 0;
}
```

Output:

3. Shortest Job First (Non Preemptive)
Code:

```c
// Author: Chaudhary Hamdan, 1905387

#include<stdio.h>

int main()
{
#ifndef ONLINE_JUDGE
        freopen("C:\\Users\\KIIT\\input", "r", stdin);
        freopen("C:\\Users\\KIIT\\output", "w", stdout);
#endif

        int i, n, p[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, min, k = 1, btime = 0;
        int bt[10], temp, j, at[10], wt[10], tt[10], ta = 0, sum = 0;
        float wavg = 0, tavg = 0, tsum = 0, wsum = 0;
        scanf("%d", &n);
        for (i = 0; i < n; i++)
        {
                scanf(" %d", &at[i]);
                scanf(" %d", &bt[i]);
        }
        for (i = 0; i < n; i++)
        {
                for (j = 0; j < n; j++)
                {
                        if (at[i] < at[j])
                        {
                                temp = p[j];
                                p[j] = p[i];
                                p[i] = temp;
                                temp = at[j];
                                at[j] = at[i];
                                at[i] = temp;
                                temp = bt[j];
                                bt[j] = bt[i];
                                bt[i] = temp;
                        }
                }
        }
```

```c
        }
        for (j = 0; j < n; j++)
        {
                btime = btime + bt[j];
                min = bt[k];
                for (i = k; i < n; i++)
                {
                        if (btime >= at[i] && bt[i] < min) {
                                temp = p[k];
                                p[k] = p[i];
                                p[i] = temp;
                                temp = at[k];
                                at[k] = at[i];
                                at[i] = temp;
                                temp = bt[k];
                                bt[k] = bt[i];
                                bt[i] = temp;
                        }
                } k++;
        }
        wt[0] = 0;
        for (i = 1; i < n; i++)
        {
                sum = sum + bt[i - 1];
                wt[i] = sum - at[i];
                wsum = wsum + wt[i];
        }
        wavg = (wsum / n);
        for (i = 0; i < n; i++)
        {
                ta = ta + bt[i];
                tt[i] = ta - at[i];
                tsum = tsum + tt[i];
        }
        tavg = (tsum / n);

        printf("Average Waiting Time: %f", wavg);
        printf("\nAverage Turnaround Time: %f", tavg);
        return 0;
}
```
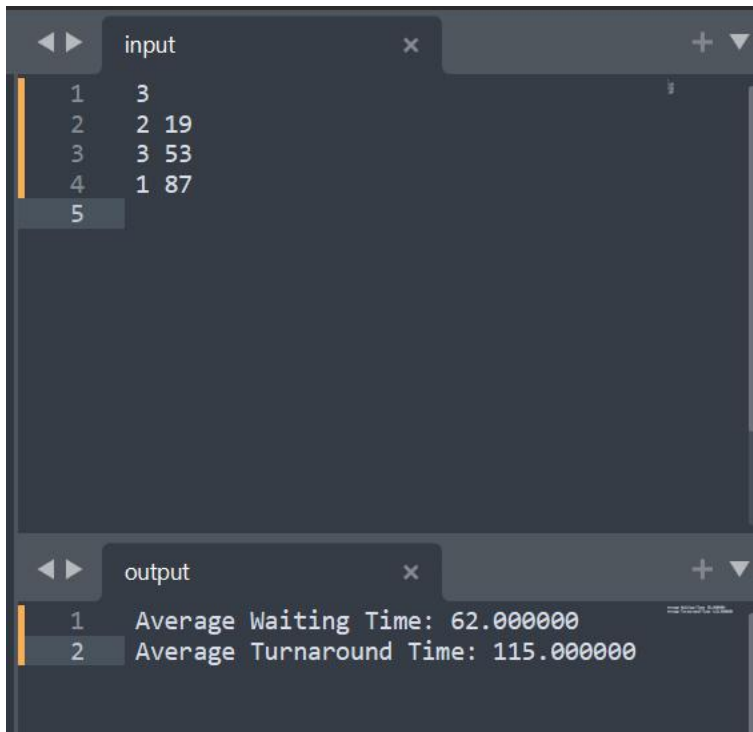
Output:

```
input                              ×              + ▼
1   3
2   2 19
3   3 53
4   1 87
5
```

```
output                             ×              + ▼
1   Average Waiting Time: 62.000000
2   Average Turnaround Time: 115.000000
```

4. Round Robin
Code:

```c
// Author: Chaudhary Hamdan, 1905387

#include<stdio.h>

int main()
{

#ifndef ONLINE_JUDGE
        freopen("C:\\Users\\KIIT\\input", "r", stdin);
        freopen("C:\\Users\\KIIT\\output", "w", stdout);
#endif



        int i, limit, total = 0, x, counter = 0, time_quantum;
        int wait_time = 0, turnaround_time = 0, arrival_time[10], burst_time
                            [10], temp[10];
        float average_wait_time, average_turnaround_time;
        scanf("%d", &limit);
        x = limit;
        for (i = 0; i < limit; i++)
        {
                scanf("%d", &arrival_time[i]);
                scanf("%d", &burst_time[i]);
                temp[i] = burst_time[i];
        }
        scanf("%d", &time_quantum);
        for (total = 0, i = 0; x != 0;)
        {
                if (temp[i] <= time_quantum && temp[i] > 0)
                {
                        total = total + temp[i];
                        temp[i] = 0;
                        counter = 1;
                }
                else if (temp[i] > 0)
                {
```

```c
                temp[i] = temp[i] - time_quantum;
                total = total + time_quantum;
        }
        if (temp[i] == 0 && counter == 1)
        {
                x--;
                wait_time = wait_time + total - arrival_time[i] -
                        burst_time[i];
                turnaround_time = turnaround_time + total -
                        arrival_time[i];
                counter = 0;
        }
        if (i == limit - 1)
        {
                i = 0;
        }
        else if (arrival_time[i + 1] <= total)
        {
                i++;
        }
        else
        {
                i = 0;
        }
    }
    average_wait_time = wait_time * 1.0 / limit;
    average_turnaround_time = turnaround_time * 1.0 / limit;
    printf("Average Waiting Time:%f", average_wait_time);
    printf("\nAvg Turnaround Time:%f\n", average_turnaround_time);
    return 0;
}
```
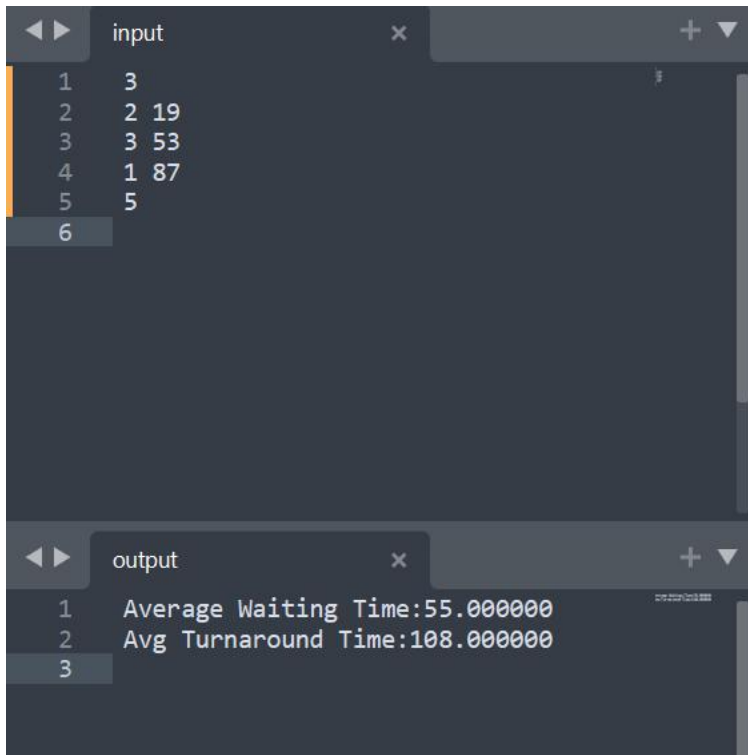
Output:

```
input                    ×           + ▼
1    3
2    2 19
3    3 53
4    1 87
5    5
6
```

```
output                   ×           + ▼
1    Average Waiting Time:55.000000
2    Avg Turnaround Time:108.000000
3
```

Average Waiting Time:

FCFS: 28
SJF Preemptive: 30
SJF Non Preemptive: 62
Round Robin: 55

Graph: