Title: CPU Scheduling
Problem Statement: Implement the C program for CPU Scheduling

Algorithms: Shortest Job First (Pre-emptive) and Round Robin with different arrival time.

//Shortest remaining time first

```c
#include <stdio.h>
int main()
{
   int AT[10], BT[10], temp[10];
   int i, smallest, count = 0, time, n;
   double WT = 0, TAT = 0, exitime;
   float average_waiting_time, average_TAT;
   printf("\nEnter the Total Number of Processes:\t");
   scanf("%d", &n);
   printf("\nEnter Details of %d Processes\n", n);
   for (i = 0; i < n; i++)
   {
       printf("\nEnter Arrival Time:\t");
       scanf("%d", &AT[i]);
       printf("Enter Burst Time:\t");
       scanf("%d", &BT[i]);
       temp[i] = BT[i];
   }
   BT[9] = 9999; // infinity value
   for (time = 0; count != n; time++)
   {
       smallest = 9;
       for (i = 0; i < n; i++)
       {
           if (AT[i] <= time && BT[i] < BT[smallest] && BT[i] > 0)
           {
               smallest = i;
           }
       }
       BT[smallest]--;
       if (BT[smallest] == 0)
       {
           count++;
           exitime = time + 1;
           WT = WT + exitime - AT[smallest] - temp[smallest];
           TAT = TAT + exitime - AT[smallest];
       }
   }
   average_TAT = TAT / n;
   average_waiting_time = WT / n;
   printf("\n\nAverage Waiting Time: \t%lf\n", average_waiting_time);
   printf("Average Turnaround Time:\t%lf\n", average_TAT);
   return 0;
}
```

```
eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$ nano shortest_Remain_Time.c
eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$ gcc -o d shortest_Remain_Time.c
eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$ ./d

Enter the Total Number of Processes:    5

Enter Details of 5 Processes

Enter Arrival Time:     10
Enter Burst Time:       5

Enter Arrival Time:     10
Enter Burst Time:       10

Enter Arrival Time:     5
Enter Burst Time:       4

Enter Arrival Time:     20
Enter Burst Time:       10

Enter Arrival Time:     25
Enter Burst Time:       15


Average Waiting Time:   4.000000
Average Turnaround Time:        12.800000
```

==========

//code for round robin

```c
#include <stdio.h>
int main()
{
   int count, j, n, exittime, remain, flag = 0, time_quantum;
   int AT[6], BT[6], RT[6];
   float WT = 0, TAT = 0;
   printf("Enter Total Process:\t");
   scanf("%d", &n);
   remain = n;

   for (count = 0; count < n; count++)
   {
       printf("Enter Arrival Time and Burst Time for Process Process Number %d: ", count + 1);
       scanf("%d", &AT[count]);
       scanf("%d", &BT[count]);
       RT[count] = BT[count]; // remaining time
   }
   printf("Enter Time Quantum:\t");
   scanf("%d", &time_quantum);
   printf("\n\nProcess\t | Turnaround Time | Waiting Time\n\n");

   for (exittime = 0, count = 0; remain != 0;)
   {
       if (RT[count] <= time_quantum && RT[count] > 0)
```

```
        {
            exittime += RT[count];
            RT[count] = 0;
            flag = 1;
        }
        else if (RT[count] > 0)
        {
            RT[count] -= time_quantum;
            exittime += time_quantum;
        }
        if (RT[count] == 0 && flag == 1)
        {
            remain--;
            printf("P[%d]\t|\t%d\t|\t%d\n", count + 1, exittime - AT[count], exittime - AT[count] -
BT[count]);
            WT += exittime - AT[count] - BT[count];
            TAT += exittime - AT[count];
            flag = 0;
        }
        if (count == n - 1)
            count = 0;
        else if (AT[count + 1] <= exittime)
            count++;
        else
            count = 0;
    }

    printf("Average Waiting Time: %f \n", WT * 1.0 / n);
    printf("Average Turnaround Time: %f", TAT * 1.0 / n);
    return 0;
}
```

```
eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$ nano roundRobin.c
eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$ gcc -o e roundRobin.c
eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$ ./e
Enter Total Process:    3
Enter Arrival Time and Burst Time for Process Process Number 1: 1 6
Enter Arrival Time and Burst Time for Process Process Number 2: 0 3
Enter Arrival Time and Burst Time for Process Process Number 3: 4 7
Enter Time Quantum:     2


Process  | Turnaround Time | Waiting Time

P[2]     |       9         |       6
P[1]     |       12        |       6
P[3]     |       12        |       5
Average Waiting Time: 5.666667
Average Turnaround Time: 11.000000eeshan@eeshan-VirtualBox:~/Desktop/Os_prac$
```