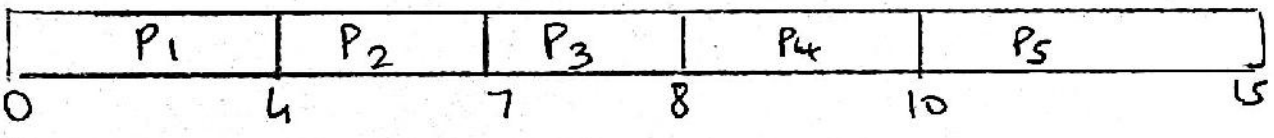
M-SATHISH KUMAR
E0119052

C PROGRAMS ON CPU Scheduling

1. First Come First Serve

Process Number	Arrival Time (AT)	Burst Time (BT)	Completion Time	Turn around Time	Weight Time
P_1	0	4	4	4	0
P_2	1	3	7	6	3
P_3	2	1	8	6	5
P_4	3	2	10	7	5
P_5	4	5	15	11	6
				34	19

Grantt Chart :



Avg TAT = 34/5

= 6.8 ms

Avg WT = 19/5

= 3.8 ms

C 1_FirstComeFirstServe.c > main()

```c
#include <stdio.h>
#include <string.h>
#include <conio.h>
main()
{
    char pn[10][10], t[10];
    int arr[10], bur[10], star[10], finish[10], tat[10], wt[10], i, j, n, temp;
    int totwt = 0, tottat = 0;
    //clrscr();
    printf("Enter the number of processes:");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {   printf("Enter the ProcessName, Arrival Time& Burst Time:");
        scanf("%s%d%d", &pn[i], &arr[i], &bur[i]);}
    for (i = 0; i < n; i++)
    {for (j = 0; j < n; j++)
        {if (arr[i] < arr[j])
            {   temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
                temp = bur[i];
                bur[i] = bur[j];
                bur[j] = temp;
                strcpy(t, pn[i]);
                strcpy(pn[i], pn[j]);
                strcpy(pn[j], t);
            }}}
    for (i = 0; i < n; i++)
    {   if (i == 0)
            star[i] = arr[i];
        else
            star[i] = finish[i - 1];
        wt[i] = star[i] - arr[i];
        finish[i] = star[i] + bur[i];
        tat[i] = finish[i] - arr[i];}
    printf("\nPName Arrtime Burtime WaitTime Start TAT Finish");
    for (i = 0; i < n; i++)
    {   printf("\n%s\t%3d\t%3d\t%3d\t%3d\t%6d\t%6d", pn[i], arr[i], bur[i], wt[i], star[i], tat[i], finish[i]);
        totwt += wt[i];
        tottat += tat[i];
    }
    printf("\nAverage Waiting time:%f", (float)totwt / n);
    printf("\nAverage Turn Around Time:%f", (float)tottat / n);
    getch();
    return 0;
}
```

E:\OpeartingSystems Programs For Practice\Tasks\DAY - 2\DAY 2\DAY 2 Bin\1_FirstComeFirstServe.exe

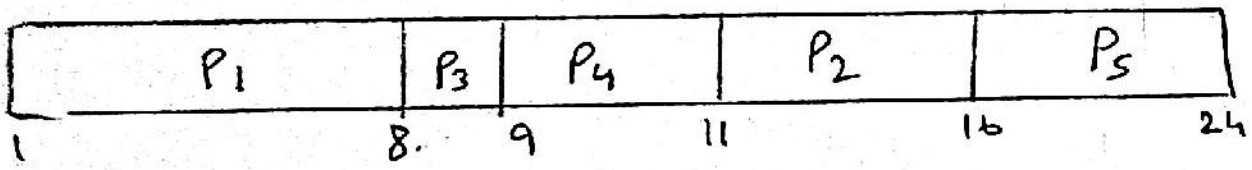
```
Enter the number of processes:5
Enter the ProcessName, Arrival Time& Burst Time:P1 0 4
Enter the ProcessName, Arrival Time& Burst Time:P2 1 3
Enter the ProcessName, Arrival Time& Burst Time:P3 2 1
Enter the ProcessName, Arrival Time& Burst Time:P4 3 2
Enter the ProcessName, Arrival Time& Burst Time:P5 4 5

PName Arrtime Burtime WaitTime Start TAT Finish
P1        0       4       0       0          4       4
P2        1       3       3       4          6       7
P3        2       1       5       7          6       8
P4        3       2       5       8          7      10
P5        4       5       6      10         11      15
Average Waiting time:3.800000
Average Turn Around Time:6.800000
```

2. SHORTEST JOB FIRST

Process Number	ARRIVAL TIME	BURST TIME	COMPLETION TIME	TURN AROUND TIME	WEIGHT TIME
P_1	1	7	8	7	0
P_2	2	5	16	14	9
P_3	3	1	9	6	5
P_4	4	2	11	7	5
P_5	5	8	24	19	11
				53	30

Grantt Chart :



Avg TAT = 53/5

= 10.6 ms

Avg WT = 30/5

= 6 ms

C 2_ShortestJobFirst.c > main()

```c
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    int et[20], at[10], n, i, j, temp, st[10], ft[10], wt[10], ta[10];
    int totwt = 0, totta = 0;
    float awt, ata;
    char pn[10][10], t[10];
    printf("Enter the number of process:");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {printf("Enter process name, arrival time& execution time:");
        //flushall();
        scanf("%s%d%d", pn[i], &at[i], &et[i]);}
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
        {   if (et[i] < et[j])
            {   temp = at[i];
                at[i] = at[j];
                at[j] = temp;
                temp = et[i];
                et[i] = et[j];
                et[j] = temp;
                strcpy(t, pn[i]);
                strcpy(pn[i], pn[j]);
                strcpy(pn[j], t);}}
    for (i = 0; i < n; i++){
        if (i == 0)
            st[i] = at[i];
        else
            st[i] = ft[i - 1];
        wt[i] = st[i] - at[i];
        ft[i] = st[i] + et[i];
        ta[i] = ft[i] - at[i];
        totwt += wt[i];
        totta += ta[i];}
    awt = (float)totwt / n;
    ata = (float)totta / n;
    printf("\nPname\tarrivaltime\texecutiontime\twaitingtime\ttatime");
    for (i = 0; i < n; i++)
        printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d", pn[i], at[i], et[i], wt[i], ta[i]);
    printf("\nAverage waiting time is:%f", awt);
    printf("\nAverage turnaroundtime is:%f", ata);
    getch();
}
```

E:\OpeartingSystems Programs For Practice\Tasks\DAY - 2\DAY 2\DAY 2 Bin\2_ShortestJobFirst.exe

```
Enter the number of process:5
Enter process name, arrival time& execution time:P1 1 7
Enter process name, arrival time& execution time:P2 2 5
Enter process name, arrival time& execution time:P3 3 1
Enter process name, arrival time& execution time:P4 4 2
Enter process name, arrival time& execution time:P5 5 8

Pname   arrivaltime     executiontime   waitingtime     tatime
P3          3               1               0               1
P4          4               2               0               2
P2          2               5               4               9
P1          1               7              10              17
P5          5               8              13              21
Average waiting time is:5.400000
Average turnaroundtime is:10.000000
```

3. SHORTEST REMAINING TIME FIRST:

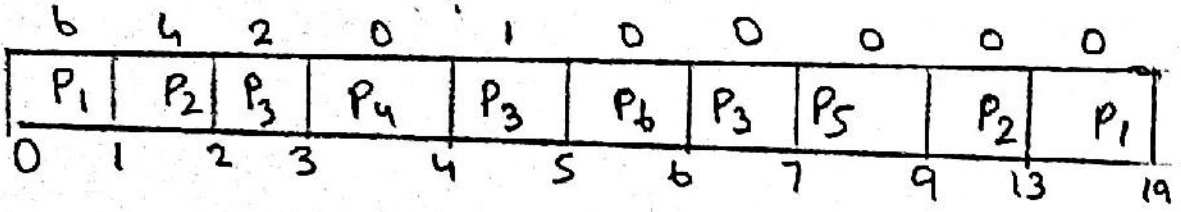
Process Number	ARRIVAL TIME	BURST TIME	COMPLETION TIME	Turn around Time	Weight Time
P_1	0	7	19	19	12
P_2	1	5	13	12	7
P_3	2	3	7	5	2
P_4	3	1	4	1	0
P_5	4	2	9	5	3
P_6	5	1	6	1	0
				43	24

Gantt chart:

Avg TAT = 43/6

= 7.116 milliseconds

Avg WT = 24/6 = 4 ms



C 3_ShortestRemainigJob.c > main()

```c
#include <stdio.h>
int main()
{
    int a[10], b[10], x[10], i, j, smallest, count = 0, time, n;
    double avg = 0, tt = 0, end;
    printf("enter the number of Processes:\n");
    scanf("%d", &n);
    printf("enter arrival time\n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    printf("enter burst time\n");
    for (i = 0; i < n; i++)
        scanf("%d", &b[i]);
    for (i = 0; i < n; i++)
        x[i] = b[i];

    b[9] = 9999;

    for (time = 0; count != n; time++)
    {
        smallest = 9;
        for (i = 0; i < n; i++)
        {
            if (a[i] <= time && b[i] < b[smallest] && b[i] > 0)
                smallest = i;
        }
        b[smallest]--;
        if (b[smallest] == 0)
        {
            count++;
            end = time + 1;
            avg = avg + end - a[smallest] - x[smallest];
            tt = tt + end - a[smallest];
        }
    }
    printf("\n\nAverage waiting time = %lf\n", avg / n);
    printf("Average Turnaround time = %lf", tt / n);
    return 0;
}
```

```
E:\OpeartingSystems Programs For Practice\Tasks\DAY - 2\DAY 2\DAY 2>.\"3_ShortestRemainigJob.exe"
enter the number of Processes:
6
enter arrival time
0
1
2
3
4
5
enter burst time
7
5
3
1
2
1


Average waiting time = 4.000000
Average Turnaround time = 7.166667
```

4.

PROCESS NUMBER	ARRIVAL TIME	BURST TIME	COMPLETION TIME	TURN AROUND TIME	WEIGHT TIME
P_1	0	4	8	8	4
P_2	1	5	18	17	12
P_3	2	2	6	4	2
P_4	3	1	9	6	5
P_5	4	6	21	17	11
P_6	6	3	19	13	10

(i) Time Quantum : 2 sec

Grantt chart:

P_1	P_2	P_3	P_1	P_4	P_5	P_2	P_6	P_5	P_2	P_6	P_5

$$\text{avg TAT} = \frac{8+17+4+6+17+13}{6}$$

$$= 10.9$$

$$\text{avg WT} = \frac{4+12+2+5+11+10}{6}$$

$$= 7.3$$

C 4_RoundRobin.c > main()

```c
#include<stdio.h>

int main()
{ int count,j,n,time,remain,flag=0,time_quantum;
  int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
  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];
  }
  printf("Enter Time Quantum:\t");
  scanf("%d",&time_quantum);
  printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
  for(time=0,count=0;remain!=0;)
  {
    if(rt[count]<=time_quantum && rt[count]>0)
    {
      time+=rt[count];
      rt[count]=0;
      flag=1;
    }
    else if(rt[count]>0)
    { rt[count]-=time_quantum;
      time+=time_quantum; }
    if(rt[count]==0 && flag==1)
    { remain--;
      printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);
      wait_time+=time-at[count]-bt[count];
      turnaround_time+=time-at[count];
      flag=0; }
    if(count==n-1)
      count=0;
    else if(at[count+1]<=time)
      count++;
    else
      count=0;}
  printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
  printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);

  return 0;
}
```

```
E:\OpeartingSystems Programs For Practice\Tasks\DAY - 2\DAY 2\DAY 2>.\"4_RoundRobin.exe"
Enter Total Process:     6
Enter Arrival Time and Burst Time for Process Process Number 1 :0 4
Enter Arrival Time and Burst Time for Process Process Number 2 :1 5
Enter Arrival Time and Burst Time for Process Process Number 3 :2 2
Enter Arrival Time and Burst Time for Process Process Number 4 :3 1
Enter Arrival Time and Burst Time for Process Process Number 5 :4 6
Enter Arrival Time and Burst Time for Process Process Number 6 :6 3
Enter Time Quantum:     2


Process |Turnaround Time|Waiting Time

P[3]    |       4       |       2
P[4]    |       4       |       3
P[1]    |       13      |       9
P[6]    |       12      |       9
P[2]    |       18      |       13
P[5]    |       17      |       11

Average Waiting Time= 7.833333
Avg Turnaround Time = 11.333333
```