

```
mmcoe02@mmcoe02:~$ cd Desktop
mmcoe02@mmcoe02:~/Desktop$ gcc sjc.c
mmcoe02@mmcoe02:~/Desktop$ ./a.out
-----Shortest Job First Scheduling ( NP )-----
```

```
Enter the No. of processes :4
Enter the burst time of 1 process :7
Enter the arrival time of 1 process :0
Enter the burst time of 2 process :4
Enter the arrival time of 2 process :2
Enter the burst time of 3 process :2
Enter the arrival time of 3 process :1
Enter the burst time of 4 process :5
Enter the arrival time of 4 process :3
```

```
*****
```

```
RESULT:-
```

Process	Burst	Arrival	Waiting	Turn-around
p1	7	0	0	7
p3	2	1	6	8
p2	4	2	7	11
p4	5	3	10	15

```
AVERAGE WAITING TIME : 5.750000
AVERAGE TURN AROUND TIME : 10.250000mmcoe02@mmcoe02:~/Desktop$
```

2)Round Robin

```
CODE:
#include<stdio.h>
#include<conio.h>

void main()
{
// initialize the variable name
```

```

int i, NOP, sum=0, count=0, y, quant, wt=0, tat=0, at[10], bt[10], temp[10];
float avg_wt, avg_tat;
printf(" Total number of process in the system: ");
scanf("%d", &NOP);
y = NOP; // Assign the number of process to variable y

// Use for loop to enter the details of the process like Arrival time and the Burst Time
for(i=0; i<NOP; i++)
{
    printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i+1);
    printf(" Arrival time is: \t"); // Accept arrival time
    scanf("%d", &at[i]);
    printf(" \nBurst time is: \t"); // Accept the Burst time
    scanf("%d", &bt[i]);
    temp[i] = bt[i]; // store the burst time in temp array
}
// Accept the Time quantum
printf("Enter the Time Quantum for the process: \t");
scanf("%d", &quant);
// Display the process No, burst time, Turn Around Time and the waiting time
printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");
for(sum=0, i = 0; y!=0; )
{
    if(temp[i] <= quant && temp[i] > 0) // define the conditions
    {
        sum = sum + temp[i];
        temp[i] = 0;
        count=1;
    }
    else if(temp[i] > 0)
    {
        temp[i] = temp[i] - quant;
        sum = sum + quant;
    }
    if(temp[i]==0 && count==1)
    {
        y--; //decrement the process no.
        printf("\nProcess No[%d] \t\t %d\t\t\t\t %d\t\t\t %d", i+1, bt[i], sum-at[i], sum-at[i]-bt[i]);
        wt = wt+sum-at[i]-bt[i];
        tat = tat+sum-at[i];
        count =0;
    }
    if(i==NOP-1)
    {
        i=0;
    }
    else if(at[i+1]<=sum)
    {
        i++;
    }
    else
    {
        i=0;
    }
}

```

```
// represents the average waiting time and Turn Around time
avg_wt = wt * 1.0/NOP;
avg_tat = tat * 1.0/NOP;
printf("\n Average Turn Around Time: \t%f", avg_wt);
printf("\n Average Waiting Time: \t%f", avg_tat);
getch();
}
```

OUTPUT:

Total number of process in the system: 5

Enter the Arrival and Burst time of the Process[1]

Arrival time is: 0

Burst time is: 10

Enter the Arrival and Burst time of the Process[2]

Arrival time is: 0

Burst time is: 6

Enter the Arrival and Burst time of the Process[3]

Arrival time is: 1

Burst time is: 7

Enter the Arrival and Burst time of the Process[4]

Arrival time is: 1

Burst time is: 4

Enter the Arrival and Burst time of the Process[5]

Arrival time is: 2

Burst time is: 5

Enter the Time Quantum for the process: 5

Process No	Burst Time	TAT	Waiting Time
Process No[4]	4	18	14

Process No[5]	5	22	17
Process No[1]	10	29	19
Process No[2]	6	30	24
Process No[3]	7	31	24

Average Turn Around Time: 19.600000

Average Waiting Time: 26.000000