

SHORTEST JOB FIRST:

```
#include <stdio.h>
int main()
{
    int A[100][4]; // Matrix for storing Process Id, Burst
                  // Time, Average Waiting Time & Average
                  // Turn Around Time.
    int i, j, n, total = 0, index, temp;
    float avg_wt, avg_tat;
    printf("Enter number of process: ");
    scanf("%d", &n);
    printf("Enter Burst Time:\n");
    // User Input Burst Time and allotting Process Id.
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &A[i][1]);
        A[i][0] = i + 1;
    }
    // Sorting process according to their Burst Time.
    for (i = 0; i < n; i++) {
        index = i;
        for (j = i + 1; j < n; j++)
            if (A[j][1] < A[index][1])
                index = j;
        temp = A[i][1];
        A[i][1] = A[index][1];
        A[index][1] = temp;

        temp = A[i][0];
        A[i][0] = A[index][0];
        A[index][0] = temp;
    }
    A[0][2] = 0;
    // Calculation of Waiting Times
    for (i = 1; i < n; i++) {
        A[i][2] = 0;
        for (j = 0; j < i; j++)
            A[i][2] += A[j][1];
    }
}
```

```

        total += A[i][2];
    }
    avg_wt = (float)total / n;
    total = 0;
    printf("P      BT      WT      TAT\n");
    // Calculation of Turn Around Time and printing the
    // data.
    for (i = 0; i < n; i++) {
        A[i][3] = A[i][1] + A[i][2];
        total += A[i][3];
        printf("P%d      %d      %d      %d\n", A[i][0],
                A[i][1], A[i][2], A[i][3]);
    }
    avg_tat = (float)total / n;
    printf("Average Waiting Time= %f", avg_wt);
    printf("\nAverage Turnaround Time= %f", avg_tat);
}

```

OUTPUT:

```

(kali㉿kali)-[~/Desktop/20CYS281]
$ ./a.out
Enter Total Process:      5
Enter Arrival Time and Burst Time for Process Process Number 1 :2
5
Enter Arrival Time and Burst Time for Process Process Number 2 :4
1
Enter Arrival Time and Burst Time for Process Process Number 3 :5
2
Enter Arrival Time and Burst Time for Process Process Number 4 :5
6
Enter Arrival Time and Burst Time for Process Process Number 5 :8
4
Enter Time Quantum:      2

Process |Turnaround Time|Waiting Time
P[2]    |      1      |      0
P[3]    |      2      |      0
P[1]    |     10      |      5
P[5]    |      8      |      4
P[4]    |     13      |      7

Average Waiting Time= 3.200000
Avg Turnaround Time = 6.800000

```

PRIORITY SCHEDULING :

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;
```

```
    printf("Enter Total Number of Process:");
```

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

```
    printf("\nEnter Burst Time and Priority\n");
```

```
    for(i=0;i<n;i++)
```

```

{
    printf("\nP[%d]\n",i+1);
    printf("Burst Time:");
    scanf("%d",&bt[i]);
    printf("Priority:");
    scanf("%d",&pr[i]);
    p[i]=i+1;    //contains process number
}

```

//sorting burst time, priority and process number in ascending order using selection sort

```
for(i=0;i<n;i++)
```

```

{
    pos=i;
    for(j=i+1;j<n;j++)
    {
        if(pr[j]<pr[pos])
            pos=j;
    }
}

```

//sorting burst time, priority and process number in ascending order using selection sort

```
for(i=0;i<n;i++)
```

```

{
    pos=i;
    for(j=i+1;j<n;j++)
    {
        if(pr[j]<pr[pos])
            pos=j;
    }
}

```

```
temp=pr[i];
```

```
pr[i]=pr[pos];
```

```
pr[pos]=temp;
```

```
temp=bt[i];
```

```
bt[i]=bt[pos];
```

```
bt[pos]=temp;
```

```
temp=p[i];
```

```
p[i]=p[pos];
```

```
p[pos]=temp;
```

```
}
```

```
wt[0]=0;
```

```
//waiting time for first process is zero
```

```
for(i=1;i<n;i++)
```

```
{
```

```
    wt[i]=0;
```

```
    for(j=0;j<i;j++)
```

```
        wt[i]+=bt[j];
```

```
    total+=wt[i];
```

```
}
```

```
avg_wt=total/n;    //average waiting time
```

```
total=0;
```

```
printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
```

```
for(i=0;i<n;i++)
```

```
{
```

```
    tat[i]=bt[i]+wt[i];    //calculate turnaround time
```

```

    total+=tat[i];

    printf("\nP[%d]\t\t %d\t\t %d\t\t %d",p[i],bt[i],wt[i],tat[i]);
}

avg_tat=total/n; //average turnaround time
printf("\n\nAverage Waiting Time=%d",avg_wt);
printf("\n\nAverage Turnaround Time=%d\n",avg_tat);

return 0;
}

```

OUTPUT:

Process	Burst Time	Waiting Time	Turnaround Time
P[4]	5	0	5
P[2]	1	5	6
P[3]	2	6	8
P[1]	2	8	10

Average Waiting Time=4
Average Turnaround Time=7