CPU scheduling algorithms

- **FCFS**

```
#include<iostream>

using namespace std;


int main()

{   int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;


        cout<<"\n \t\tFirst Come First Serve Scheduling Algorithm\n\n"<<endl;

        cout<<"Enter total number of processes:";

        cin>>n;


        cout<<"\nEnter Process Burst Time -  \n";

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

        {

      cout<<"P["<<i+1<<"]:";

        cin>>bt[i];

        }

        wt[0]=0;

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

        {
```

```
        wt[i]=0;
        for(j=0;j<i;j++)
        wt[i]+=bt[j];
        }
        cout<<"\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time";
        for(i=0;i<n;i++)
        {
        tat[i]=bt[i]+wt[i];
        avwt+=wt[i];
        avtat+=tat[i];
      cout<<"\nP["<<i+1<<"]"<<"\t\t"<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<tat[i];
        }
        avwt/=i;
        avtat/=i;
   cout<<"\n\nAverage Waiting Time:\n"<<avwt;
   cout<<"\n"<<endl;
        cout<<"\nAverage Turnaround Time:\n"<<avtat;
   cout<<"\n"<<endl;

        return 0;
}
```

Output:

```
Process         Burst Time      Waiting Time    Turnaround Time
P[1]            8               0               8
P[2]            4               8               12
P[3]            9               12              21
P[4]            5               21              26

Average Waiting Time:
10


Average Turnaround Time:
16
```

- **Priority**

```
#include<iostream>
 using namespace std;
int main()
{
        int a[10],b[10],x[10];
        int waiting[10],turnaround[10],completion[10],p[10];
        int i,j,smallest,count=0,time,n;
        double avg=0,tt=0,end;

        cout<<"\n\n\t\t Priority Based Scheduling Algorithm\n"<<endl;

   cout<<"\nEnter the number of Processes: ";
        cin>>n;
```

```
for(i=0;i<n;i++)
{
cout<<"\nEnter arrival time of process: ";
cin>>a[i];
}
for(i=0;i<n;i++)
{
cout<<"\nEnter burst time of process: ";
cin>>b[i];
}
for(i=0;i<n;i++)
{
cout<<"\nEnter priority of process: ";
cin>>p[i];
}
for(i=0; i<n; i++)
x[i]=b[i];

p[9]=-1;
for(time=0; count!=n; time++)
{
smallest=9;
for(i=0; i<n; i++)
```

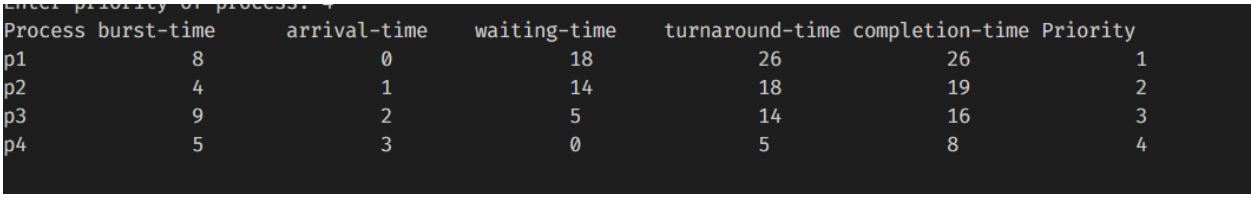
```
        {
        if(a[i]<=time && p[i]>p[smallest] && b[i]>0 )
                smallest=i;
        }
        b[smallest]--;

        if(b[smallest]==0)
        {
        count++;
        end=time+1;
        completion[smallest] = end;
        waiting[smallest] = end - a[smallest] - x[smallest];
        turnaround[smallest] = end - a[smallest];
        }
        }
    cout<<"Process"<<"\t"<< "burst-time"<<"\t"<<"arrival-time" <<"\t"<<"waiting-time"
<<"\t"<<"turnaround-time"<< "\t"<<"completion-time"<<"\t"<<"Priority"<<endl;
        for(i=0; i<n; i++)
        {

        cout<<"p"<<i+1<<"\t\t"<<x[i]<<"\t\t"<<a[i]<<"\t\t"<<waiting[i]<<"\t\t"<<turnaround[i]
<<"\t\t"<<completion[i]<<"\t\t"<<p[i]<<endl;
        avg = avg + waiting[i];
        tt = tt + turnaround[i];
        }
```

```
   cout<<"\n\nAverage waiting time ="<<avg/n;
        cout<<"  Average Turnaround time ="<<tt/n<<endl;
}
```

Output:



```
Enter priority of process: 4
Process burst-time      arrival-time    waiting-time    turnaround-time completion-time Priority
p1              8               0               18              26              26              1
p2              4               1               14              18              19              2
p3              9               2               5               14              16              3
p4              5               3               0               5               8               4
```

- ## Shortest Job First (SJF)

```
#include<iostream>
using namespace std;



int main()
{
        int n,temp,tt=0,min,d,i,j;
        float atat=0,awt=0,stat=0,swt=0;

    cout<<"\n\t\tShortest JOb First Scheduling Algorithm\n"<<endl;

        cout<<"Enter total number of processes : "<<endl;
        cin>>n;
        int a[n],b[n],e[n],tat[n],wt[n];
```

```
for(i=0;i<n;i++)
{
cout<<"Enter Arrival Time - ";
cin>>a[i];
}

for(i=0;i<n;i++)
{
cout<<"Enter Burst Time - ";
cin>>b[i];
}

for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
        if(b[i]>b[j])
        {
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
```

```
            temp=b[i];
            b[i]=b[j];
            b[j]=temp;
            }
    }
    }

    min=a[0];
for(i=0;i<n;i++)
    {
    if(min>a[i])
    {
            min=a[i];
            d=i;
    }
    }

    tt=min;
    e[d]=tt+b[d];
    tt=e[d];

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

```
if(a[i]!=min)
    {
        e[i]=b[i]+tt;
        tt=e[i];
}
}

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

tat[i]=e[i]-a[i];
stat=stat+tat[i];
wt[i]=tat[i]-b[i];
swt=swt+wt[i];
}

atat=stat/n;
awt=swt/n;

cout<<"Process  Arrival-time(s)  Burst-time(s)  Waiting-time(s)  Turnaround-time(s)\n";

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

```
        {
   cout<<"P"<<i+1<<"          "<<a[i]<<"          "<<b[i]<<"          "<<wt[i]<<"
"<<tat[i]<<endl;
        }

        cout<<"Average Waiting Time="<<awt << endl;
        cout<<" Average Turn Around Time="<<atat << endl;
}
```

Output:

```
Process  Arrival-time(s)  Burst-time(s)  Waiting-time(s)  Turnaround-time(s)
P1              1               4                7               11
P2              3               5                9               14
P3              0               8                0               8
P4              2               9                15               24
Average Waiting Time=7.75
 Average Turn Around Time=14.25
```

Round Robin:

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

void main()
{

        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("\n\t\tRound Robin Scheduling Algorithm \n\n");
        printf(" Total number of process in the system: ");
        scanf("%d", &NOP);
        y = NOP; // Assign the number of process to variable y


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
}


printf("Enter the Time Quantum for the process: \t");
scanf("%d", &quant);


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;
        }
}

avg_wt = wt * 1.0/NOP;
avg_tat = tat * 1.0/NOP;

printf("\n\n Average Turn Around Time: \t%f\n", avg_wt);
printf("\n\n Average Waiting Time: \t%f\n", avg_tat);
printf("\n");
```

```
getch();

}
```

Output:

```
Enter the Time Quantum for the process:         1

 Process No              Burst Time              TAT             Waiting Time
Process No[2]            4                               13                      9
Process No[4]            5                               16                      11
Process No[1]            8                               24                      16
Process No[3]            9                               24                      15

 Average Turn Around Time:      12.750000


 Average Waiting Time:  19.250000
```