

CPU SCHEDULING ALGORITHMS

1. First Come First Serve

```
// CPU Scheduling- FCFS
#include<stdio.h>
// Struct Storing all the Processes
struct process
{
    int process;
    int arrival;
    int burst;
    int wait;
    int response;
    int tat;
}p[10];

// Main Function
int main()
{
    int n,awwt=0,avtat=0,i,j,pos,max,temp,sum=0;
    printf("\nEnter total number of processes:");
    scanf("%d",&n);

    // Taking the Input from the User
    for(i=0;i<n;i++)
    {
        p[i].process = i+1;
        printf("\n Process [%d] : \n",i+1);
        printf("\n Arrival Time:");
        scanf("%d",&p[i].arrival);
        printf("\n Burst Time:");
        scanf("%d",&p[i].burst);
    }

    // Sorting all the Processes according to their
    Arrival Time

    for(i=0;i<n;i++)
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(p[j].arrival<p[pos].arrival)
            {
                pos=j;
            }
        }

        temp=p[i].arrival;
        p[i].arrival=p[pos].arrival;
        p[pos].arrival=temp;
```

```
temp=p[i].burst;
p[i].burst=p[pos].burst;
p[pos].burst=temp;

temp=p[i].process;
p[i].process=p[pos].process;
p[pos].process=temp;
}
p[0].wait = 0;
// Calculating Waiting and Response Time
for(i=1;i<n;i++)
{
    sum = sum + p[i-1].burst;
    p[i].wait = sum - p[i].arrival ;
    if(p[i].wait <0)
        p[i].wait= 0;
    p[i].response = p[i].wait;
}

printf("\nProcess\t\tArrival Time\tBurst
Time\tResponse Time\tWaiting
Time\tTurnaround Time");

// Calculating Turn Around Time and Printing
the Table
for(i=0;i<n;i++)
{
    p[i].tat=p[i].burst+p[i].wait;
    awwt+=p[i].wait;
    avtat+=p[i].tat;
    printf("\nP[%d]\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t",
    p[i].process,p[i].arrival,p[i].burst,p[i].response,p
    [i].wait,p[i].tat);
}

awwt/=i;
avtat/=i;
printf("\n\nAverage Waiting Time:%d",awwt);
printf("\nAverage Turnaround Time:
%d\n\n",avtat);

return 0;
}
```