first come, first served (FCFS)ALGORITHM

AIM:

PROGRAM TO IMPLEMENT first come, first served using c language.

First Come, First Served (FCFS) also known as First In, First Out(FIFO) is the CPU scheduling algorithm in which the CPU is allocated to the processes in the order they are queued in the ready queue.

FCFS follows non-preemptive scheduling which mean once the CPU is allocated to a process it does not leave the CPU until the process will not get terminated or may get halted due to some I/O interrupt.

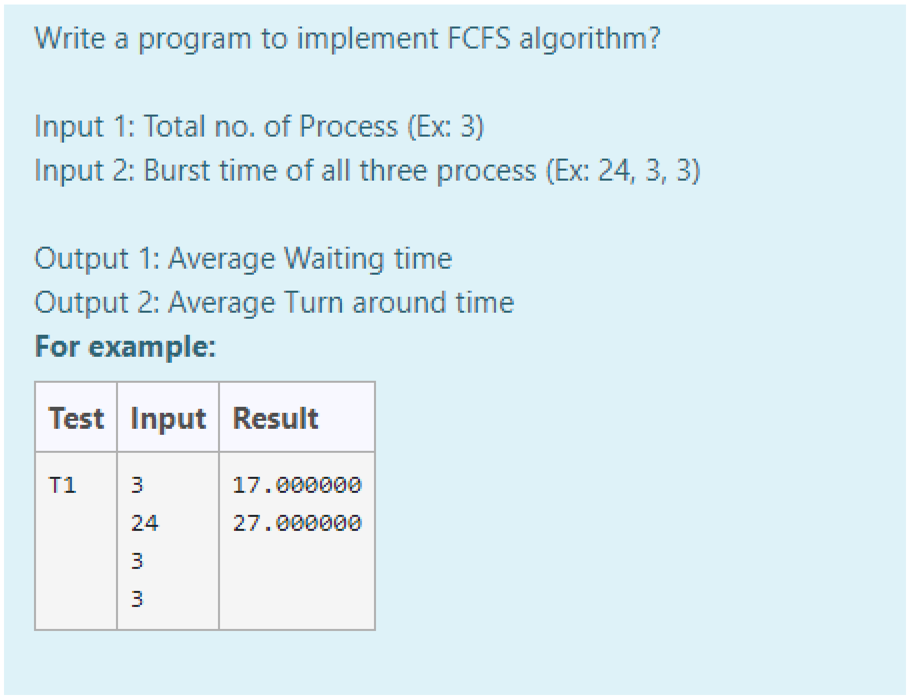
ALGORITHM:

```
1-  Input the processes along with their burst time (bt).
2-  Find waiting time (wt) for all processes.
3-  As first process that comes need not to wait so
    waiting time for process 1 will be 0 i.e. wt[0] = 0.
4-  Find waiting time for all other processes i.e. for
     process i ->
       wt[i] = bt[i-1] + wt[i-1] .
5-  Find turnaround time = waiting_time + burst_time
    for all processes.
6-  Find average waiting time =
                 total_waiting_time / no_of_processes.
7-  Similarly, find average turnaround time =
                 total_turn_around_time / no_of_processes.
```

OR

ALGORITHM:

```
Start

Step 1-> In function int waitingtime(int proc[], int n, int
burst_time[], int wait_time[])
   Set wait_time[0] = 0
   Loop For i = 1 and i < n and i++
      Set wait_time[i] = burst_time[i-1] + wait_time[i-1]
   End For
Step 2-> In function int turnaroundtime( int proc[], int n, int
burst_time[], int wait_time[], int tat[])
   Loop For  i = 0 and i < n and i++
      Set tat[i] = burst_time[i] + wait_time[i]
   End For
Step 3-> In function int avgtime( int proc[], int n, int
burst_time[])
   Declare and initialize wait_time[n], tat[n], total_wt = 0,
total_tat = 0;
   Call waitingtime(proc, n, burst_time, wait_time)
   Call turnaroundtime(proc, n, burst_time, wait_time, tat)
   Loop For  i=0 and i<n and i++
      Set total_wt = total_wt + wait_time[i]
      Set total_tat = total_tat + tat[i]
      Print process number, burstime wait time and turnaround time
   End For
   Print "Average waiting time =i.e. total_wt / n
   Print "Average turn around time = i.e. total_tat / n
Step 4-> In int main()
   Declare the input int proc[] = { 1, 2, 3}
   Declare and initialize n = sizeof proc / sizeof proc[0]
   Declare and initialize burst_time[] = {10, 5, 8}
   Call avgtime(proc, n, burst_time)
Stop
```

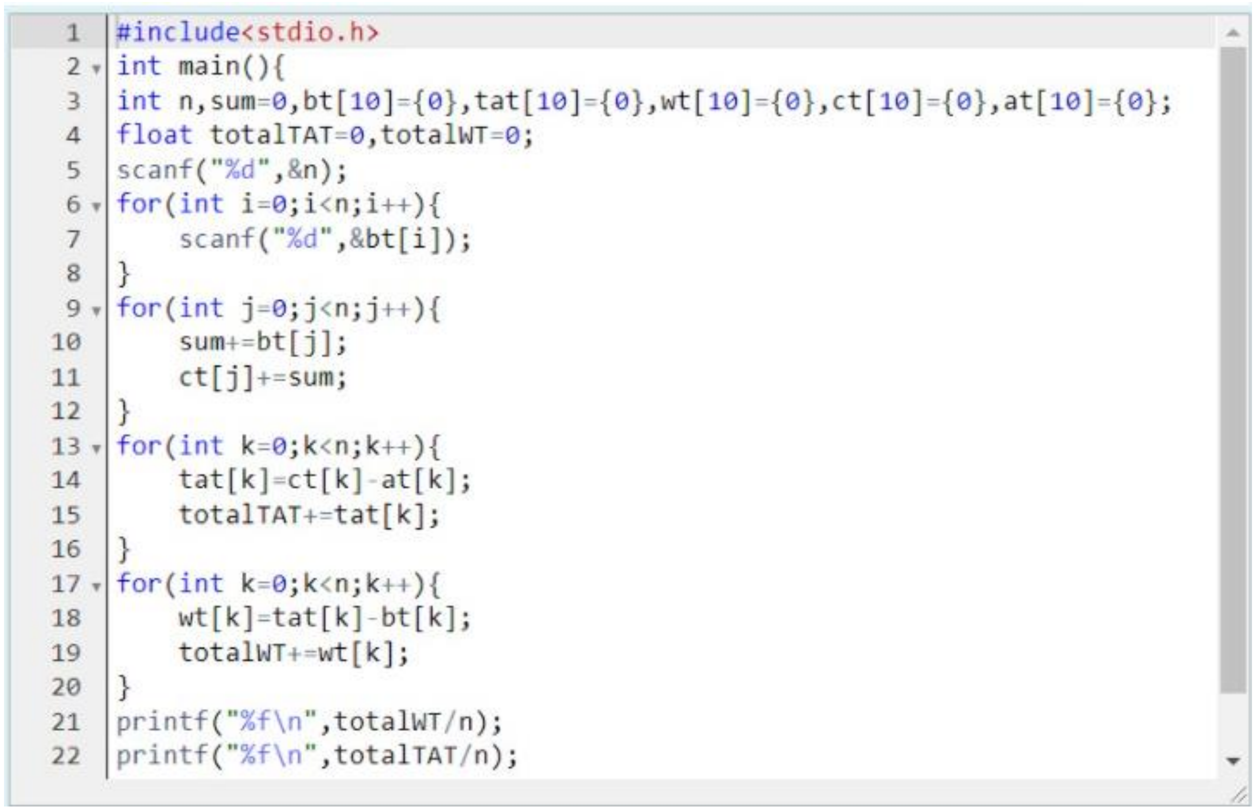

Write a program to implement FCFS algorithm?

Input 1: Total no. of Process (Ex: 3)
Input 2: Burst time of all three process (Ex: 24, 3, 3)

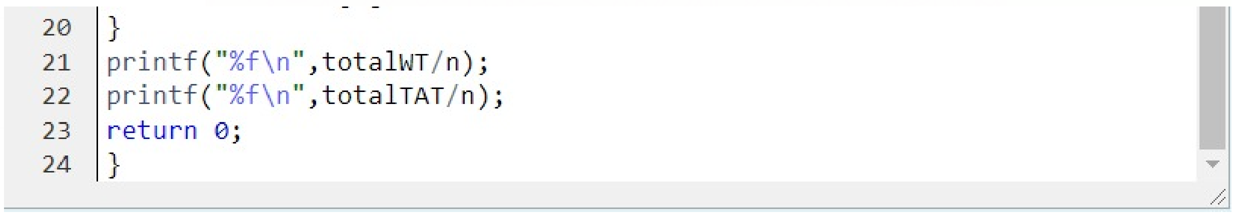
Output 1: Average Waiting time
Output 2: Average Turn around time
For example:

Test	Input	Result
T1	3 24 3 3	17.000000 27.000000

PROGRAM:



```c
#include<stdio.h>
int main(){
int n,sum=0,bt[10]={0},tat[10]={0},wt[10]={0},ct[10]={0},at[10]={0};
float totalTAT=0,totalWT=0;
scanf("%d",&n);
for(int i=0;i<n;i++){
    scanf("%d",&bt[i]);
}
for(int j=0;j<n;j++){
    sum+=bt[j];
    ct[j]+=sum;
}
for(int k=0;k<n;k++){
    tat[k]=ct[k]-at[k];
    totalTAT+=tat[k];
}
for(int k=0;k<n;k++){
    wt[k]=tat[k]-bt[k];
    totalWT+=wt[k];
}
printf("%f\n",totalWT/n);
printf("%f\n",totalTAT/n);
```



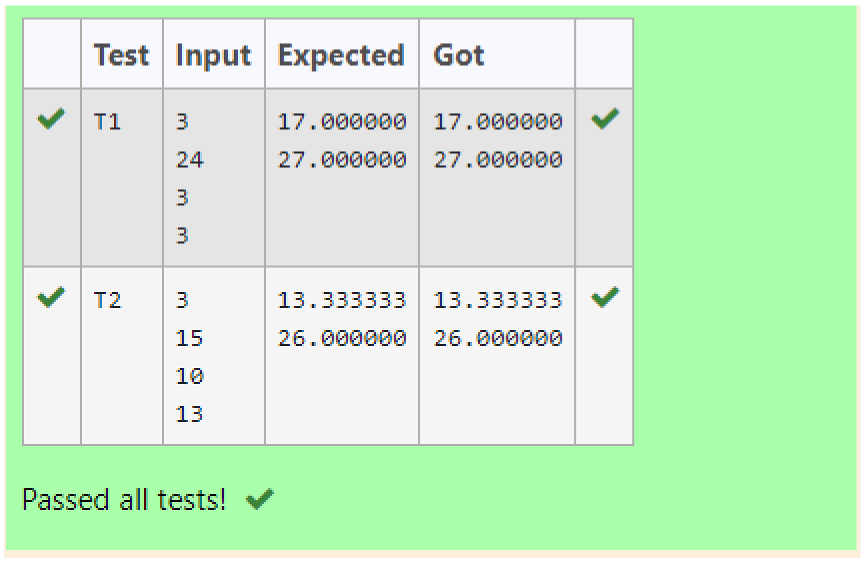
```
}
printf("%f\n",totalWT/n);
printf("%f\n",totalTAT/n);
return 0;
}
```

CODE:

```
#include<stdio.h>

int main()

{

   int n,sum=0,bt[10]={0},tat[10]={0},wt[10]={0},at[10]={0},ct[10]={0};

   float totalTAT=0,totalWT=0;

   scanf("%d",&n);

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

      scanf("%d",&bt[i]);

   }

   for(int j=0;j<n;j++){

      sum+=bt[j];

      ct[j]+=sum;

   }

   for(int k=0;k<n;k++){

      tat[k]=ct[k]-at[k];

      totalTAT+=tat[k];
```

```
    }
    for(int k=0;k<n;k++){
        wt[k]=tat[k]-bt[k];
        totalWT+=wt[k];
    }
    printf("%f\n",totalWT/n);
    printf("%f\n",totalTAT/n);
    return 0;
}
```

RESULT:



	Test	Input	Expected	Got	
✓	T1	3 24 3 3	17.000000 27.000000	17.000000 27.000000	✓
✓	T2	3 15 10 13	13.333333 26.000000	13.333333 26.000000	✓

Passed all tests! ✓

RESULT:

FCFS PROGRAM WAS IMPLEMENT SUCCESSFULLY USING C LANGUAGE.