S.No: 5	Exp. Name: ***Implement CPU Scheduling Algorithms***	**Date: 2022-04-26**

Aim:

Write a program to implement the FCFS process scheduling algorithm.

Source Code:

os.c

```
#include <stdio.h>
#include <conio.h>
#include <curses.h>
void Process_sort(int a[],int b[], int p[],int n)
{
   int key1, key2, key3, i;
   for(int j = 1; j<n; j++)
   {
      key1 = a[j];
      key2 = b[j];
      key3 = p[j];
      i = j-1;
      while (i >= 0 && a[i] > key1)
      {
         a[i+1] = a[i];
         b[i+1] = b[i];
         p[i+1] = p[i];
         i--;
      }
      a[i+1] = key1;
      b[i+1] = key2;
      p[i+1] = key3;
   }
}
int main()
{
   int n;
   float Avg_wt = 0;
   float Avg_tat = 0;
   printf("Enter the number of processes: ");
   scanf("%d", &n);
   int pn[n], at[n], bt[n], start[n], wt[n], tat[n], finish[n];
   for (int i=0; i<n; i++)
   {
      printf("Enter the Process Name, Arrival Time & Burst Time:");
      //taking the input from the user for process name
      //arrival time and burst time
      scanf("%d %d %d", &pn[i], &at[i], &bt[i]);
   }
   Process_sort(at,bt,pn,n);
   printf("Process Name\tArrival Time\tBurst Time\n");
   for (int i = 0; i<n; i++)
   {
      //printing out the values stored in the given array
      printf("    %d\t        %d\t        %d\n", pn[i], at[i], bt[i]);
```

Page No:

ID: 2001330130175

2020-24-IT-C2

Noida Institute of Engineering and Technology

```
      if (i == 0 || at[i] > finish[i-1])
      {
         start[i] = at[i];
         finish[i] = start[i] + bt[i];
         tat[i] = finish[i] - at[i];
         wt[i] = tat[i] - bt[i];
      }
      else
      {
         start[i] = finish[i-1];
         finish[i] = start[i] + bt[i];
         tat[i] = finish[i] - at[i];
         wt[i] = tat[i] - bt[i];
      }
   }
    printf("PName     Arrtime     Bursttime    Start     WT   TAT   Finish\n");
     for(int i=0; i<n; i++)
      {
         if (i == 0)
         printf("%d\t  %d\t\t  %d\t  %d\t   %d\t  %d\t  %d\n",pn[i], at[i], bt[i], star
t
         [i], wt[i], tat[i], finish[i]);
         else
         printf("%d\t  %d\t\t  %d\t  %d\t  %d\t  %d\t  %d\n",pn[i], at[i],bt[i], start
[i],
         wt[i], tat[i], finish[i]);
          }
           for (int i=0; i<n; i++)
            {
               Avg_wt = Avg_wt + wt[i];
               Avg_tat = Avg_tat + tat[i];
            }
             Avg_wt = Avg_wt / n;
              Avg_tat = Avg_tat / n;
               printf("Average Waiting time:%.6f\n", Avg_wt);
                printf("Average Turn Around Time:%.6f", Avg_tat);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
Enter the number of processes:  2
Enter the Process Name, Arrival Time & Burst Time: 1 24 27
Enter the Process Name, Arrival Time & Burst Time: 1 26 27
Process Name    Arrival Time    Burst Time
    1            24              27
    1            26              27
PName     Arrtime     Bursttime    Start     WT   TAT   Finish
1         24              27       24        0     27     51
1         26              27       51       25     52     78
Average Waiting time:12.500000
```

Test Case - 1

```
Average Turn Around Time:39.500000
```

Page No:

ID: 2001330130175

2020-24-IT-C2

Noida Institute of Engineering and Technology