Ex. No.: 6a)
Date: 19/2/25

FIRST COME FIRST SERVE

Aim:

To implement First-come First- serve (FCFS) scheduling technique

Algorithm:

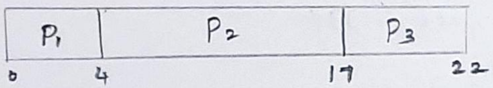
1. Get the number of processes from the user.
2. Read the process name and burst time.
3. Calculate the total process time.
4. Calculate the total waiting time and total turnaround time for each process 5. Display the process name & burst time for each process. 6. Display the total waiting time, average waiting time, turnaround time

Program Code:

```
#include <stdio.h>
int main()
{
 int n;
 printf("Enter the no. of processes: ");
 scanf("%d", &n)
int process[n], bt[n], at[n], wt[n], tat[n], ct[n];
float total_wt =0, total_tat =0;
printf("Enter the Burst time and arrival time :\n");
for (int i=1; i<=n; i++)
{
 printf("Process %d Burst Time : ", i);
 scanf("%d", &bt[i]);
}
for(int i=1; i<=n; i++)
{ printf("Process %d arrival time : ", i);
  scanf("%d", &at[i]);
}
```

```
for(int i = 1; i<n; i++){
    for(int j = 1; j<n-i; j++){
        if (at[j] > at[j+1]){
            int temp;
            temp = at[j];
           at[j] = at[j+1];
            at[j+1] = temp;
           temp = bt[j];
           bt[j] = bt[j+1];
          bt[j+1] = temp;
        }
  } }
   ct[1] = bt[1];
   for (int i = 2; i<=n; i++)
   { ct[i] = bt[i] + ct[i-1];
   }
   for (int i=1; i<=n; i++)
   { tat [i] = ct[i] - at[i];
     total_tat += tat[i];
   }
 for(int i = 1; i<=n; i++
  { wt [i] = tat[i] - bt[i];
    total_wt +=wt[i];
 }
 float avg_wt = total_wt /n;
 float avg_tat = total_tat /n;
 printf ("Process: Arrival Time: Burst Time: Completion time: Turn Around Time: Waiting Time \n");
  for (int i=1; i<=n; i++)
  {printf("%d  % d  %.d  %d  %d  %d \n"}; (i+1),
    ~~for (int i=1; i<=n; i++)~~ at[i], bt[i], ct[i], tat [i], wt [i]);
    {~~printf ("~~ %d }
    printf ("Average waiting time = %.2f\n", avg_wt);
   print("Average turn around time = %.2f\n; avg_tat );
   return 0;
}
```

Gantt Chart:



Tabulation:

Process	AT(ms)	BT(ms)	CT(ms)	TAT = CT - AT (ms)	WT = TAT - BT (ms)
1	0	4	4	4	0
2	0	13	17	17	4
3	0	5	22	22	17

Sample Output:

```
Enter the number of process:
3
Enter the burst time of the processes:
24 3 3

Process      Burst Time     Waiting Time   Turn Around Time

   0            24              0              24

   1             3             24              27

   2             3             27              30

Average waiting time is:  17.0
Average Turn around Time is:  19.0
```

```
Enter the no. of process : 3
Enter the burst time :
4
13
5
Process        Burst Time        Waiting Time        Turn Around Time
                                                          4
  1                4                 0
                                                          17
  2                13                4                    22
  3                5                 17

Avg. Waiting time is 7 ms
Avg. turn around time is : 14.33 ms
```

Result:

Thus the implementation First come First Serve has been successfully executed.