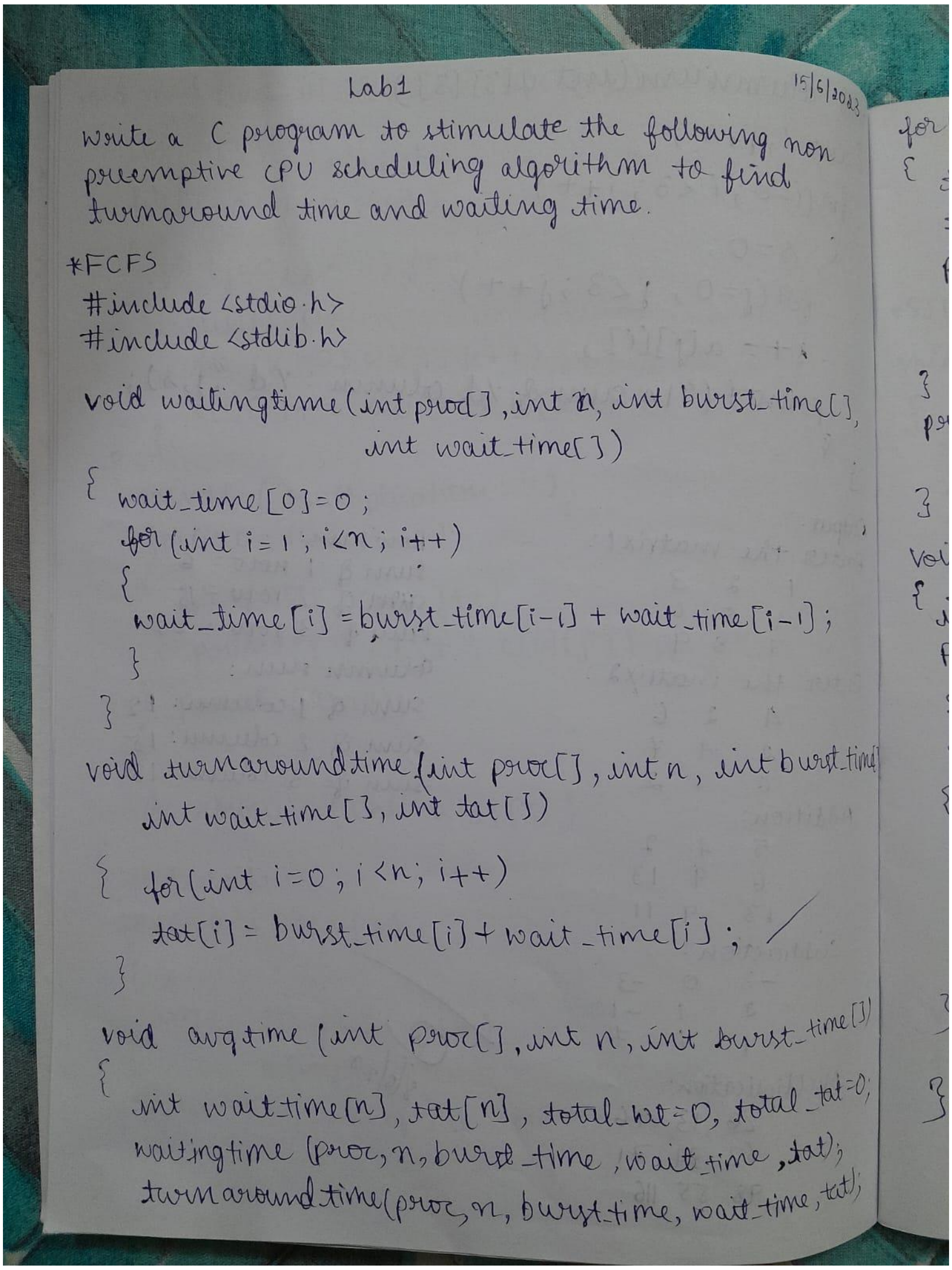
WEEK 2

Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time. *FCFS *SJF (pre-emptive & Non-pre-emptive)



Lab1

15/6/2023

write a C program to stimulate the following non preemptive CPU scheduling algorithm to find turnaround time and waiting time.

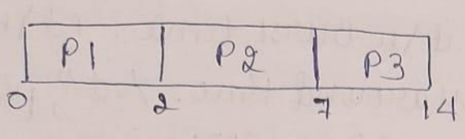
*FCFS

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

 void waitingtime(int proc[],int n, int burst_time[],
                        int wait_time[])
 {
   wait_time[0]=0;
   for (int i=1; i<n; i++)
   {
    wait_time[i] =burst_time[i-1] + wait_time[i-1];
   }
 }
 void turnaroundtime(int proc[], int n, int burst_time[]
     int wait_time[], int tat[])
 { for(int i=0; i<n; i++)
     tat[i]= burst_time[i]+ wait_time[i];
   }
  void avgtime (int proc[], int n, int burst_time[])
  {
    int wait_time[n], tat[n], total_wt=0, total_tat=0;
    waitingtime (proc, n, burst_time, wait_time, tat);
    turnaroundtime(proc, n, burst_time, wait_time, tat);
```

```
for(int i=0; i<n; i++)
{ total_wt+ = wait_time[i];
  total_tat + = tat[i];
  printf ("\n Process : %d\nBursttime: %d\n
          Waittime: %d\n Turnaroundtime: %d",
          proc[i], bursttime[i], wait time[i], tat[i]);
}
printf ("\n Average wait time: %d\n Average turnaro
       -und time: %d", total_wt/n, total_tat/n);
}
void main()
{ int proc[10], bursttime[10], n;
  Printf ("\n Enter the size of n: ");
  scanf ("%d", &n);
  for(int i=0; i<n; i++)
  { printf ("\n Enter the processor number:");
    scanf ("%d", &proc[i]);
    printf ("\nEnter the burst time:");
    scanf ("%d", &burst time[i]);
  } avgtime (proc, n, burst_time);
}
```

```
Output: Enter the size of n: 3
Enter processor number: 1
Enter burst time: 2
Enter the processor number: 2
Enter the burst time: 5
Enter the processor number: 3
Enter the burst time: 7

Process: 1
Burst Time: 2
Wait Time: 0
Turnaround time: 2
Process: 2
Burst Time: 5
Wait Time: 2
Turnaround time: 7
Process: 3
Burst Time: 7
Wait time: 7
Turnaround time: 14
Average wait time: 3
Average turnaround time: 7
```

Gaint chart :



*SJF

```
void avgtime (int proc[], int n, int burst_time[])
{
  int wait_time[n] , tat[n] , wt=0, tat=0, K;
  for(int i=0; i<n ; i++)
  { for (int j=i+1 ; j<n ; j++)
    { if(burst_time[j] < burst_time[i])
      { K=burst_time[i] ;
        burst_time[i]=burst_time[j];
        burst_time[j]=K;
        K=proc[i] ;
        proc[i]=proc[j] ;
        proc[j] =K ;
      }
    }
waiting time (proc,n,burst_time, wait_time);
turnaroundtime(proc, n,burst_time, wait_time, tat);
for (int i=0 ; i<n; i++)
{ wt+= wait_time[i];
tat+ = tat[i];
printf("\n Process: %d \n Burst time: %d\n
  wait time: %d \n Turnaround time: %d", proc[i],
  burst_time[i], wait_time[i], tat[i]);
}
printf("\n Average wait time: %d \n Average
    turnaround time: %d", wt, tat);
}
```

Output:

```
Enter the no of processor: n
Enter the processor number: 1
Enter the burst time: 10
Enter the processor number: 2
Enter the burst time: 5
Enter the processor number: 3
Enter the burst time: 8

Process: 2
Burst time: 5
Wait time: 0
Turnaround time: 5

Process: 3
Burst time: 8
Wait time: 5
Turnaround time: 13
```

Process : 1

Burst time : 10

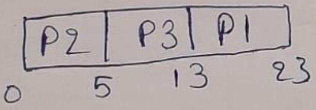
wait time : 13

Turnaround time : 23

Average wait time : 6

Average turnaround time : 13

Gaant chart :



5/.

15/6/23

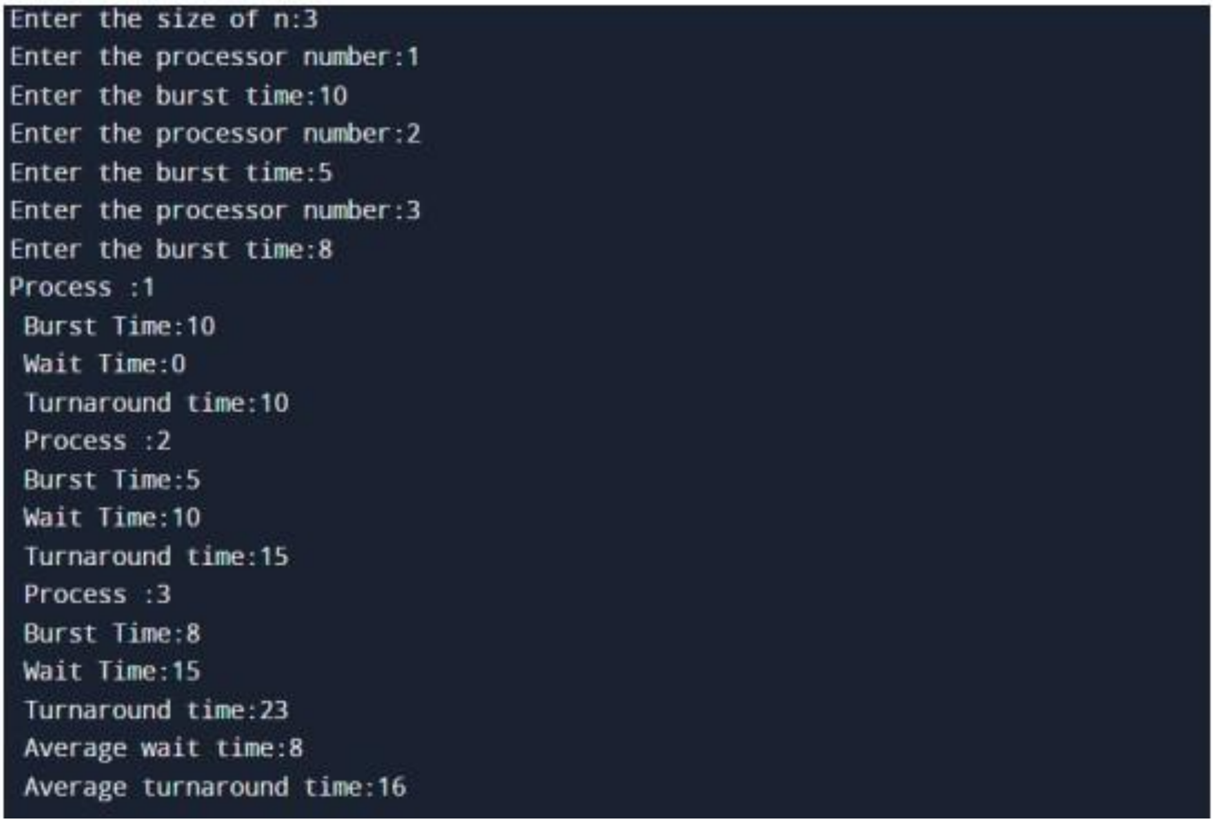
FCFS C Program:

```
#include<stdio.h>

#include<stdlib.h>

void waitingtime(int proc[],int n,int burst_time[],int wait_time[])

{

wait_time[0]=0;

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

{

wait_time[i]=burst_time[i-1]+wait_time[i-1];

}

}

void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[])

{

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

tat[i]=burst_time[i]+wait_time[i];

}

void avgtime(int proc[],int n,int burst_time[])

{
```

```
int wait_time[n],tat[n],total_wt=0,total_tat=0;
waitingtime(proc,n,burst_time,wait_time);
turnaroundtime(proc,n,burst_time,wait_time,tat);
for(int i=0;i<n;i++)
{
total_wt+=wait_time[i];
total_tat+=tat[i];
printf("\n Process :%d \n Burst Time:%d \n Wait
Time:%d \n Turnaround
time:%d",proc[i],burst_time[i],wait_time[i],tat[i]);
}
printf("\n Average wait time:%d \n Average
turnaround time:%d",total_wt/n,total_tat/n);
}
void main()
{
int proc[10],burst_time[10],n;
printf("\n Enter the size of n:");
scanf("%d",&n);
for(int i=0;i<n;i++)
{
```

```
printf("\n Enter the processor number:");
scanf("%d",&proc[i]);
printf("\n Enter the burst time:");
scanf("%d",&burst_time[i]);
}
avgtime(proc,n,burst_time);
}
```

OUTPUT:



```
Enter the size of n:3
Enter the processor number:1
Enter the burst time:10
Enter the processor number:2
Enter the burst time:5
Enter the processor number:3
Enter the burst time:8
Process :1
 Burst Time:10
 Wait Time:0
 Turnaround time:10
 Process :2
 Burst Time:5
 Wait Time:10
 Turnaround time:15
 Process :3
 Burst Time:8
 Wait Time:15
 Turnaround time:23
 Average wait time:8
 Average turnaround time:16
```

SJF C Program:

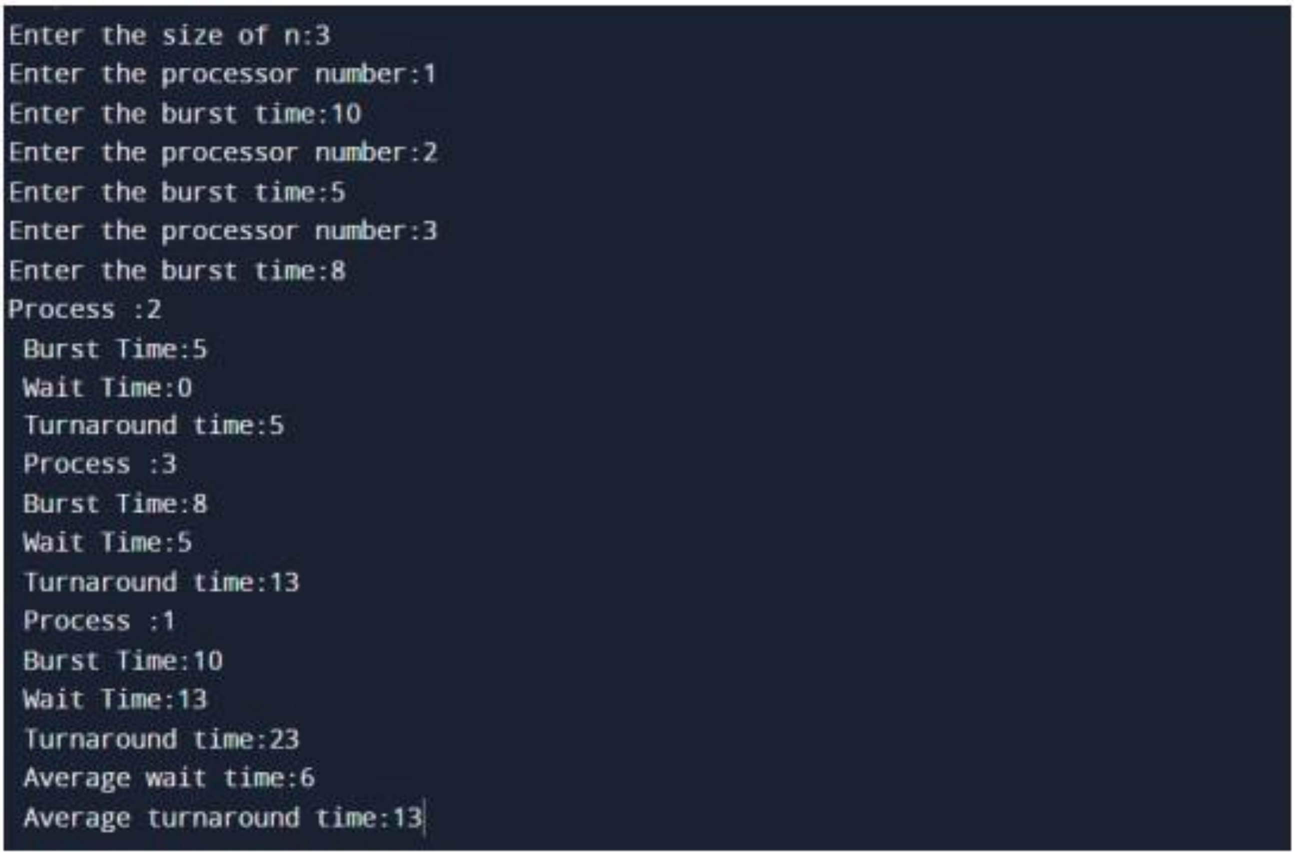
```
#include<stdio.h>

#include<stdlib.h>

void waitingtime(int proc[],int n,int burst_time[],int
wait_time[])

{

wait_time[0]=0;

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

{

wait_time[i]=burst_time[i-1]+wait_time[i-1];

}

}

void turnaroundtime(int proc[],int n,int
burst_time[],int wait_time[],int tat[])

{

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

tat[i]=burst_time[i]+wait_time[i];

}

void avgtime(int proc[],int n,int burst_time[])

{
```

```
int wait_time[n],tat[n],total_wt=0,total_tat=0,k;
for(int i=0;i<n;i++)
{
for(int j=i+1;j<n;j++)
{
 if(burst_time[j]<burst_time[i])
 {
 k=burst_time[i];
 burst_time[i]=burst_time[j];
 burst_time[j]=k;
 k=proc[i];
 proc[i]=proc[j];
 proc[j]=k;
 }
}
}
waitingtime(proc,n,burst_time,wait_time);
turnaroundtime(proc,n,burst_time,wait_time,tat);
for(int i=0;i<n;i++)
{
```

```
total_wt+=wait_time[i];
total_tat+=tat[i];
printf("\n Process :%d \n Burst Time:%d \n Wait
Time:%d \n Turnaround
time:%d",proc[i],burst_time[i],wait_time[i],tat[i]);
}
printf("\n Average wait time:%d \n Average
turnaround time:%d",total_wt/n,total_tat/n);
}
void main()
{
int proc[10],burst_time[10],n;
printf("\n Enter the size of n:");
scanf("%d",&n);
for(int i=0;i<n;i++)
{
printf("\n Enter the processor number:");
scanf("%d",&proc[i]);
printf("\n Enter the burst time:");
scanf("%d",&burst_time[i]);
}
```

```
avgtime(proc,n,burst_time);
}
```

OUTPUT:



```
Enter the size of n:3
Enter the processor number:1
Enter the burst time:10
Enter the processor number:2
Enter the burst time:5
Enter the processor number:3
Enter the burst time:8
Process :2
 Burst Time:5
 Wait Time:0
 Turnaround time:5
 Process :3
 Burst Time:8
 Wait Time:5
 Turnaround time:13
 Process :1
 Burst Time:10
 Wait Time:13
 Turnaround time:23
 Average wait time:6
 Average turnaround time:13
```