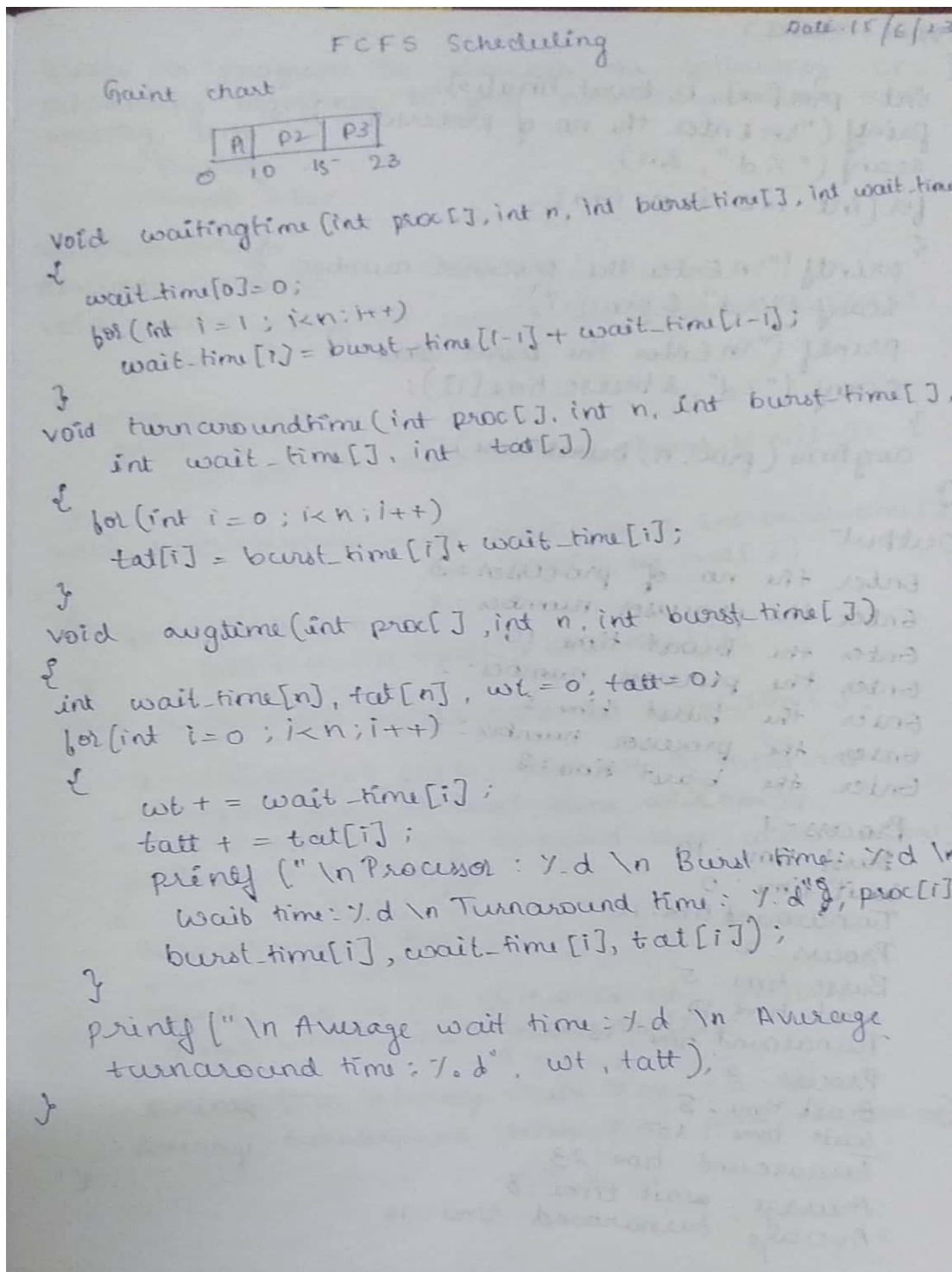


WEEK 2

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)



```
void main()
```

```
{  
    int proc[10], n, burst_time[10];  
    printf("Enter the no. of processor:");  
    scanf("%d", &n);  
    for (int i = 0; i < n; i++)  
    {  
        printf("Enter the processor number:");  
        scanf("%d", &proc[i]);  
        printf("Enter the burst time:");  
        scanf("%d", &burst_time[i]);  
    }  
    avgtime(proc, n, burst_time);  
}
```

Output

Enter the no. of processor: 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

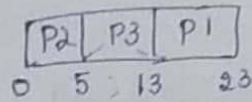
Average wait time: 8

Average turnaround time: 16

Date: 15/6/23

SJF scheduling

Gantt-chart



```

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], wt = 0, tat = 0;
    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++)
{
    wt += wait-time[i];
    tat += tat[i];
    printf("\n Process: %d \n Burst time: %d \n\n",
        proc[i], burst-time[i], wait-time[i], tat[i]);
}
printf("\n Average wait time: %d \n Average
turnaround time: %d", wt, tat);
}

void main()
{
    int proc[10], n, burst-time[10];
    printf("\n Enter the no. of processors:");
    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 no. of processor: 1

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

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
```