WEEK 2

Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time.

- Round Robin
- Priority

ROUND ROBIN

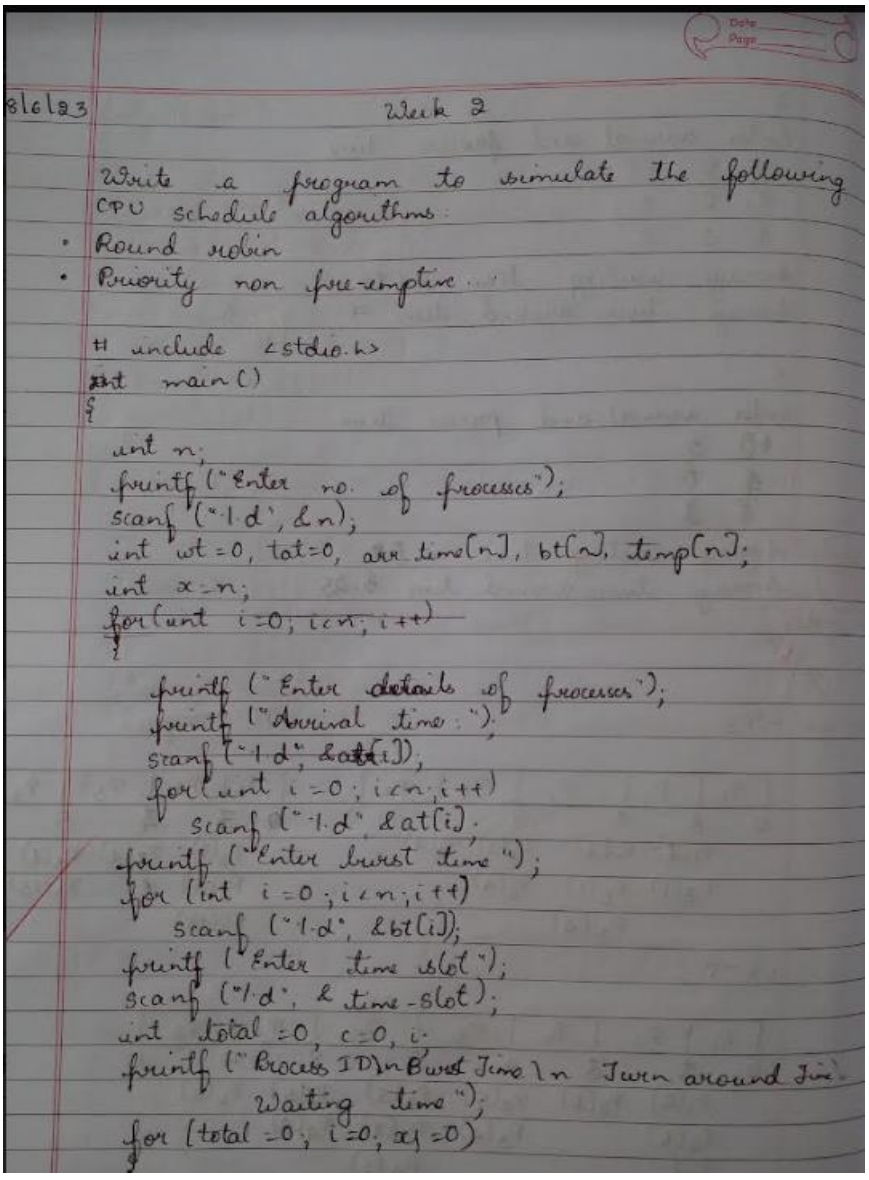
CODE:

```
#include<stdio.h>
int main()
{

   int  n;
   printf("Enter Total Number of Processes:");
   scanf("%d", &n);
   int wt = 0, tat = 0, at[n], bt[n], temp[n];
   int x = n;


   for(int i = 0; i < n; i++)
   {
      printf("Enter Details of Process %d \n", i + 1);
      printf("Arrival Time:  ");
      scanf("%d", &at[i]);
      printf("Burst Time:   ");
      scanf("%d", &bt[i]);
      temp[i] = bt[i];
   }
   int time_slot;
```

```
   printf("Enter Time Slot:");
   scanf("%d", &time_slot);
   int total = 0,  c = 0,i;
   printf("Process ID      Burst Time      Turnaround Time      Waiting
Time\n");
   for(total=0, i = 0; x!=0; )
   {

      if(temp[i] <= time_slot && temp[i] > 0)
      {
         total = total + temp[i];
         temp[i] = 0;
         c=1;
      }
      else if(temp[i] > 0)
      {
         temp[i] = temp[i] - time_slot;
         total  += time_slot;
      }
      if(temp[i]==0 && c==1)
      {
         x--;
         printf("\nProcess No %d  \t\t %d\t\t\t\t %d\t\t\t %d", i+1,
bt[i],
               total-at[i], total-at[i]-bt[i]);
        wt =wt+total-at[i]-bt[i];
         tat += total -at[i];
         c =0;
      }
      if(i==n-1)
      {
         i=0;
      }
      else if(at[i+1]<=total)
```

```
        i++;
      else
        i=0;
    }
    float average_wait_time =wt * 1.0 / n;
    float average_turnaround_time = tat * 1.0 / n;
    printf("\nAverage Waiting Time:%f", average_wait_time);
    printf("\nAvg Turnaround Time:%f", average_turnaround_time);
}
```

Observation book :



8/6/23 Week 2

Write a program to simulate the following CPU schedule algorithms:

- Round robin
- Priority non pre-emptive

```
#include <stdio.h>
int main()
{
  int n;
  printf("Enter no. of processes");
  scanf("%d", &n);
  int wt=0, tat=0, arr_time[n], bt[n], temp[n];
  int x=n;
  for(int i=0; i<n; i++)
  {
    printf("Enter details of processes");
    printf("Arrival time:");
    scanf("%d", &at[i]);
    for(int i=0; i<n; i++)
      scanf("%d", &at[i]);
    printf("Enter burst time");
    for(int i=0; i<n; i++)
      scanf("%d", &bt[i]);
    printf("Enter time slot");
    scanf("%d", &time_slot);
    int total=0, c=0, i;
    printf("Process ID\nBurst Time\n Turn around Time
          Waiting time");
    for(total=0; i=0; x!=0)
```

```
if (temp[i] <= time_slot && temp[i]>0)
{
    total = total + temp[i];
    temp[i]=0;
    c=1;
}
else if (temp[i]>0)
{
    temp[i] = temp[i]-time_slot;
    total += time_slot;
}
if (temp[i] ==0 && c==1)
{
    x--;
    printf("%d \t %d \t %d \t %d \t %d, i+1, bt[i],
        total-at[i], total-at[i]-bt[i]);
    wt = wt+ total-at[i]-bt[i];
    tat += total-at[i];
    c=0;
}
if (i==n-1)
    i=0;
else if (at[i+1]<=total)
    i++;
else
    i=0;
}
float avgwait_time = wt/n;
float avgtat_time = tat/n;
printf("Avergge waiting time : %f", avgwait_t
printf("Average turnaround time : %f", avgtat_t
}
```

Output:

```
Enter the total number of processes : 6
Enter details of processes.
Enter arrival time
0  1  2  3  4  4
Enter burst time
7  4  15  11  20  9
Enter time slot.
5
```

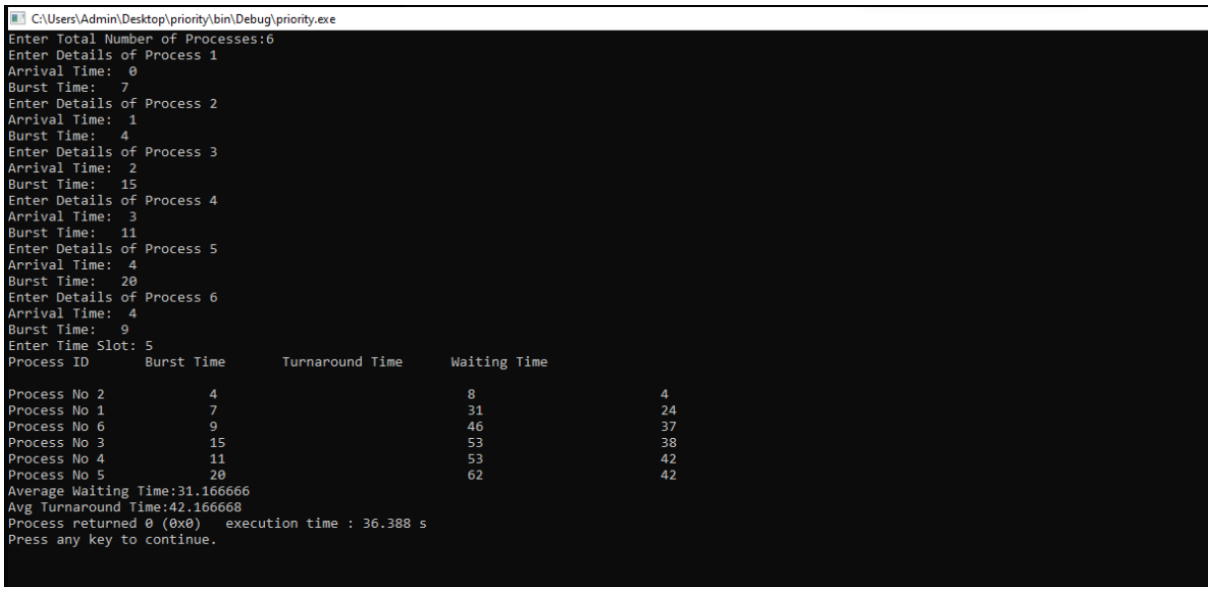
Process ID	Burst Time	Turnaround Time	Waiting Time
P2	4	8	4
P1	7	31	24
P6	9	46	37
P3	15	53	38
P4	11	53	42
P5	20	62	42

Average waiting time : 31.166
Average turnaround time : 42.166

Gantt chart

P1	P2	P3	P4	P5	P6	P1	P3	P4	P5	P6	P3	P4	P5
0	5	9	14	19	24	29	31	36	41	46	50	55	56

OUTPUT :


C:\Users\Admin\Desktop\priority\bin\Debug\priority.exe
Enter Total Number of Processes:6
Enter Details of Process 1
Arrival Time: 0
Burst Time: 7
Enter Details of Process 2
Arrival Time: 1
Burst Time: 4
Enter Details of Process 3
Arrival Time: 2
Burst Time: 15
Enter Details of Process 4
Arrival Time: 3
Burst Time: 11
Enter Details of Process 5
Arrival Time: 4
Burst Time: 20
Enter Details of Process 6
Arrival Time: 4
Burst Time: 9
Enter Time Slot: 5
Process ID Burst Time Turnaround Time Waiting Time
Process No 2 4 8 4
Process No 1 7 31 24
Process No 6 9 46 37
Process No 3 15 53 38
Process No 4 11 53 42
Process No 5 20 62 42
Average Waiting Time:31.166666
Avg Turnaround Time:42.166668
Process returned 0 (0x0) execution time : 36.388 s
Press any key to continue.

PRIORITY (NON PREEMPTIVE)

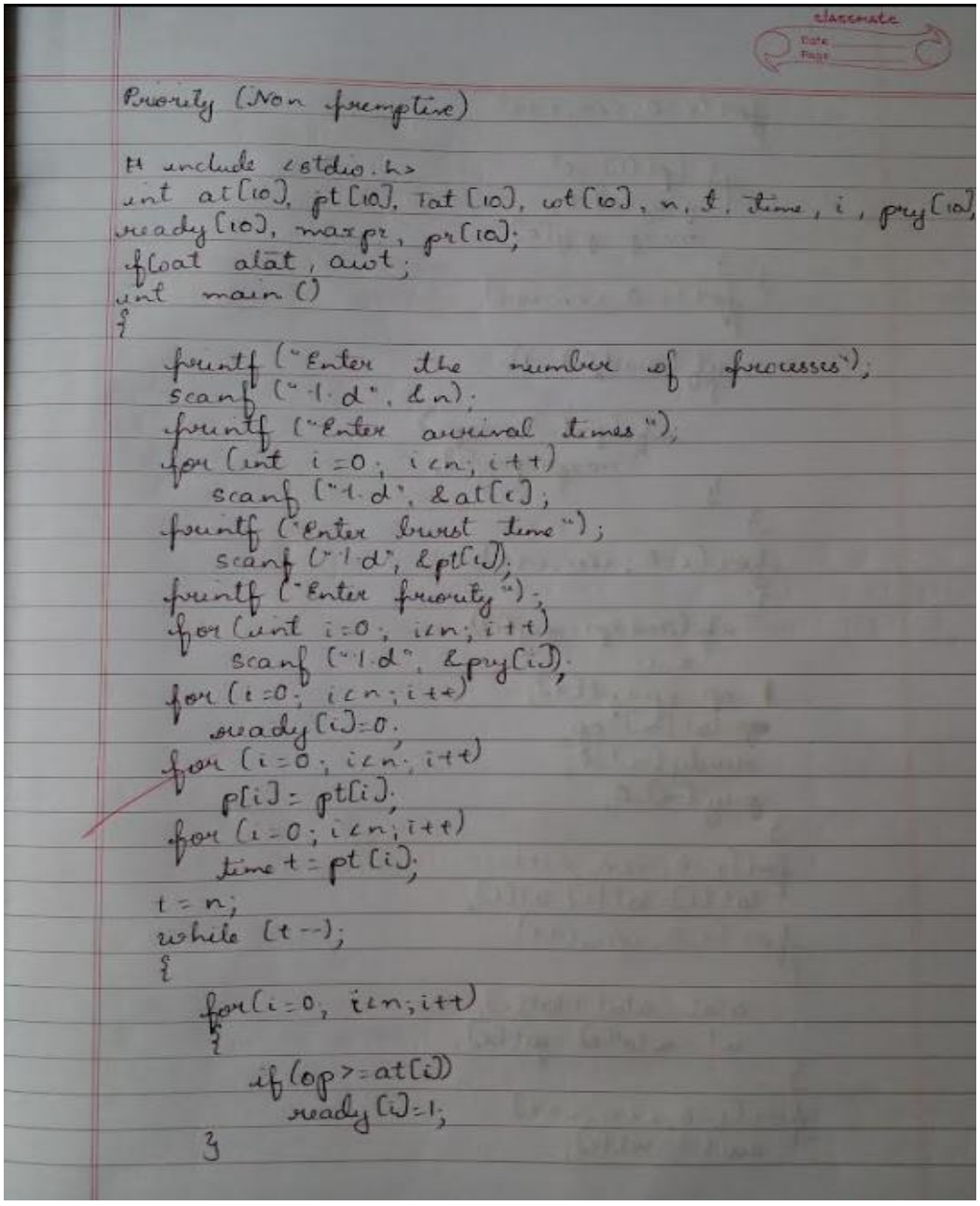
CODE:

```
#include<stdio.h>
int at[10],t,pt[10],tat[10],wt[10],n,time=0,i,ready[10],pry[10],op=0,
maxpr,x,p[10];
float atat=0,awt=0;
void main()
{
    printf("Enter number of processes \n");
    scanf("%d",&n);
   printf("Enter arrival times: \n");
    for(i=0;i<n;i++)
    scanf("%d",&at[i]);
    printf("Enter process times: \n");
    for(i=0;i<n;i++)
    scanf("%d",&pt[i]);
   printf("Enter priority: \n");
   for(i=0;i<n;i++)
    scanf("%d",&pry[i]);
    for(i=0;i<n;i++)
    ready[i]=0;
    for(i=0;i<n;i++)
```

```
p[i]=pt[i];
for(i=0;i<n;i++)
time+=pt[i];
t=n;
while(t--)
{
   for(i=0;i<n;i++)
   if(op>=at[i])
   ready[i]=1;
   for(i=0;i<n;i++)
   if(pt[i]==0)
   pry[i]=0;
   maxpr=pry[0];
   for(i=0;i<n;i++)
   if(ready[i]==1)
   if(pry[i]>maxpr)
   maxpr=pry[i];
   for(i=0;i<n;i++)
   if(maxpr==pry[i])
   x=i;
   op=op+pt[x];
   tat[x]=op;
   ready[x]=0;
   pry[x]=0;
```

```
}
for(i=0;i<n;i++)
        tat[i]=tat[i]-at[i];
    for(i=0;i<n;i++)
    {
        atat+=tat[i];
        wt[i]=tat[i]-pt[i];
    }
    for(i=0;i<n;i++)
    awt+=wt[i];
    awt=awt/n;
    atat=atat/n;
    printf("\n");
    for(i=0;i<n;i++)
    printf("P%d  %d  %d \n",(i+1),tat[i],wt[i]);
    printf("ATAT=%f  \nAWT=%f ",atat,awt);
}
```

Observation book :



Priority (Non preemptive)

```
#include <stdio.h>
int at[10], pt[10], tat[10], wt[10], n, t, time, i, pry[10],
ready[10], maxpr, pr[10];
float alat, awt;
int main()
{
    printf("Enter the number of processes");
    scanf("%d", &n);
    printf("Enter arrival times");
    for(int i=0; i<n; i++)
        scanf("%d", &at[i]);
    printf("Enter burst time");
        scanf("%d", &pt[i]);
    printf("Enter priority");
    for(int i=0; i<n; i++)
        scanf("%d", &pry[i]);
    for(i=0; i<n; i++)
        ready[i]=0;
    for(i=0; i<n; i++)
        p[i]=pt[i];
    for(i=0; i<n; i++)
        time+=pt[i];
    t=n;
    while(t--);
    {
        for(i=0, i<n; i++)
        {
            if(op>=at[i])
                ready[i]=1;
        }
```

```
for (i=0; i<n; i++)
{
    if (pt[i]==0)
        pry[i]=0;
        maxpr=pry[0];
}
for (i=0; i<n; i++)
{
    if (ready[i]==1)
        if (pw[i]>maxpr)
            maxpr=pry[i];
}
for (i=0; i<n; i++)
{
    if (maxpr==pry[i])
        x=i;
    op=op+pt[x];
    tat[x]=op;
    ready[x]=0;
    pry[x]=0;
}
for (i=0; i<n; i++)
    tat[i]=tat[i]-at[i];
for (i=0; i<n; i++)
{
    atat=atat+tat[i];
    wt=tat[i]-pt[i];
}
for (i=0; i<n; i++)
    awt+=wt[i];
```

```
    awt=awt/n;
    atat=atat/n;
    for (i=0; i<n; i++)
        printf ("P %d \t %d \t %d \t %d \n", (i+1),
                tat[i], wt[i]);
    printf ("Average waiting time : %f", awt);
    printf ("Average turn around time : %f", atat);
}
```

Output:

```
Enter number of processes : 4
Enter arrival times : 0 1 2 3
Enter process times : 4 3 3 5
Enter priority : 3 4 6 5
```

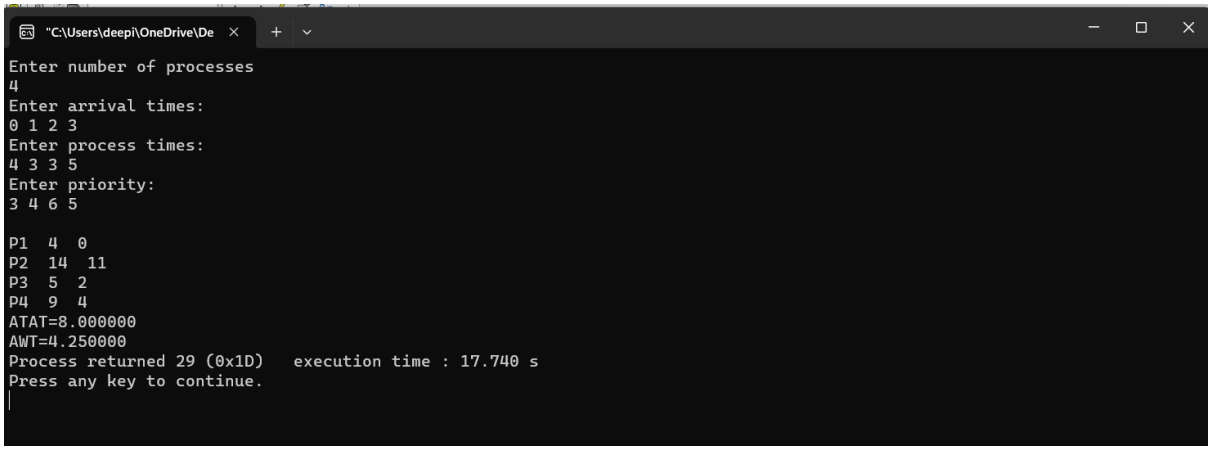
Process	Turnaround time	Waiting time
P1	4	0
P2	14	11
P3	5	2
P4	9	4

Average waiting time : 4.25
Average turnaround time : 8.00

Gantt chart

P1	P1	P1	P3	P4	P2
0	1	2	4	7	12	15

OUTPUT:


"C:\Users\deepi\OneDrive\De
Enter number of processes
4
Enter arrival times:
0 1 2 3
Enter process times:
4 3 3 5
Enter priority:
3 4 6 5
P1 4 0
P2 14 11
P3 5 2
P4 9 4
ATAT=8.000000
AWT=4.250000
Process returned 29 (0x1D) execution time : 17.740 s
Press any key to continue.