FCFS

```
#include<stdio.h>
#include<stdlib.h>
int cmpt[30],bt[30],at[30],tat[30],wat[30];
int n,i,sum;
void fcfs(int n,int bt[n],int at[n]){
    float avg_tat=0,avg_wat=0;
    for(i=0;i<n;i++) {
        sum+=bt[i];
        cmpt[i]=sum;
    }
    for(i=0;i<n;i++){
        tat[i]=cmpt[i]-at[i];
        wat[i]=tat[i]-bt[i];
    }
    printf("\n");
    for(i=0;i<n;i++){
        avg_tat=avg_tat+tat[i];
        avg_wat=avg_wat+wat[i];
    }
    avg_tat=avg_tat/n;
    avg_wat=avg_wat/n;
    printf("PROCESS\t ARRIVAL TIME\t BURST TIME\t TURN AROUND TIME\t WAITING
TIME\n");
    for(i=0;i<n;i++){
        printf("p%d\t|\t %d\t|\t %d\t|\t %d\t|\t %d\n",(i+1),at[i],bt[i],tat[i],wat[i]);
    }
    printf("\n");
    printf("Avg waiting time is:%f\n",avg_wat);
    printf("Avg turnaround time is:%f\n",avg_tat);
}
void main(){
    printf("Enter number of processes:\n");
    scanf("%d",&n);
    if(n==0)
        printf("there are no processes in queue\n ");
    else{
        printf("Enter the arrival and burst time of the processes respectively:\n");
        for(i=0;i<n;i++){
            printf("P%d\t",(i+1));
            scanf("%d%d",&at[i],&bt[i]);
        }
    }
    fcfs(n,bt,at);
}
```

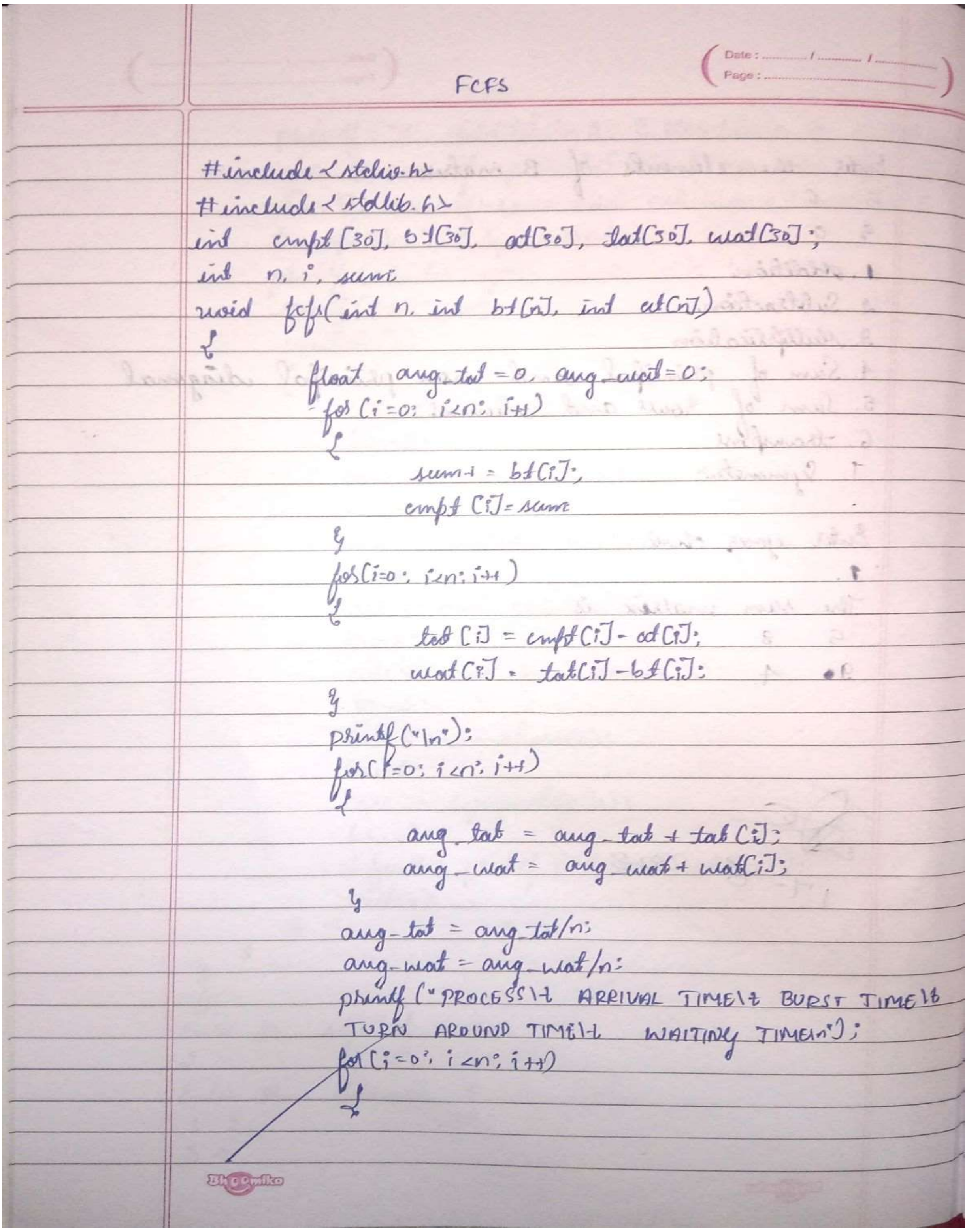
SHORTEST JOB FIRST

```
#include<stdio.h>
int main() {
  int time, bt[10], at[10], sum_bt = 0, smallest, n, i;
  int sumt = 0, sumw = 0;
  printf("enter the no of processes : ");
  scanf("%d", & n);
  printf("Enter the arrival and burst time of the processes respectively:\n");
        for(i=0;i<n;i++) {
            printf("P%d\t",(i+1));
            scanf("%d%d",&at[i],&bt[i]);
            sum_bt += bt[i];
  }
  bt[9] = 9999;
  for (time = 0; time < sum_bt;) {
  smallest = 9;
  for (i = 0; i < n; i++) {
      if (at[i] <= time && bt[i] > 0 && bt[i] < bt[smallest])
        smallest = i;
    }
    printf("P%d\t|\t%d\t|\t%d\n", smallest + 1, time + bt[smallest] - at[smallest], time - at[smallest]);
    sumt += time + bt[smallest] - at[smallest];
    sumw += time - at[smallest];
    time += bt[smallest];
    bt[smallest] = 0;
  }
  printf("\n\n average waiting time = %f", sumw * 1.0 / n);
  printf("\n\n average turnaround time = %f", sumt * 1.0 / n);
  return 0;
}
```

SHORTEST REMAINING TIME FIRST

```
#include<stdio.h>
#define MAX 9999
struct proc{
    int no,at,bt,rt,ct,tat,wt;
};
struct proc read(int i){
    struct proc p;
    printf("\nProcess No: %d\n",i);
    p.no=i;
    printf("Enter Arrival Time: ");
    scanf("%d",&p.at);
    printf("Enter Burst Time: ");
    scanf("%d",&p.bt);
    p.rt=p.bt;
    return p;
}
int main(){
    struct proc p[10],temp;
    float avgtat=0,avgwt=0;
    int n,s,remain=0,time;
    printf("Enter Number of Processes: ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        p[i]=read(i+1);
    for(int i=0;i<n-1;i++)
        for(int j=0;j<n-i-1;j++)
            if(p[j].at>p[j+1].at){
            temp=p[j];
            p[j]=p[j+1];
            p[j+1]=temp;
            }
    printf("\nProcess\t\tAT\tBT\tCT\tTAT\tWT\n");
    p[9].rt=MAX;
    for(time=0;remain!=n;time++){
        s=9;
        for(int i=0;i<n;i++)
            if(p[i].at<=time&&p[i].rt<p[s].rt&&p[i].rt>0)
                s=i;
        p[s].rt--;
        if(p[s].rt==0){
            remain++;
            p[s].ct=time+1;
            p[s].tat=p[s].ct-p[s].at;
            avgtat+=p[s].tat;
            p[s].wt=p[s].tat-p[s].bt;
            avgwt+=p[s].wt;
            printf("P%d\t\t%d\t%d\t%d\t%d\t%d\n",p[s].no,p[s].at,p[s].bt,p[s].ct,p[s].tat,p[s].wt);
        }
    }
    avgtat/=n,avgwt/=n;
    printf("\nAverage TurnAroundTime=%f\nAverage WaitingTime=%f",avgtat,avgwt);
}
```

OBSERVATION

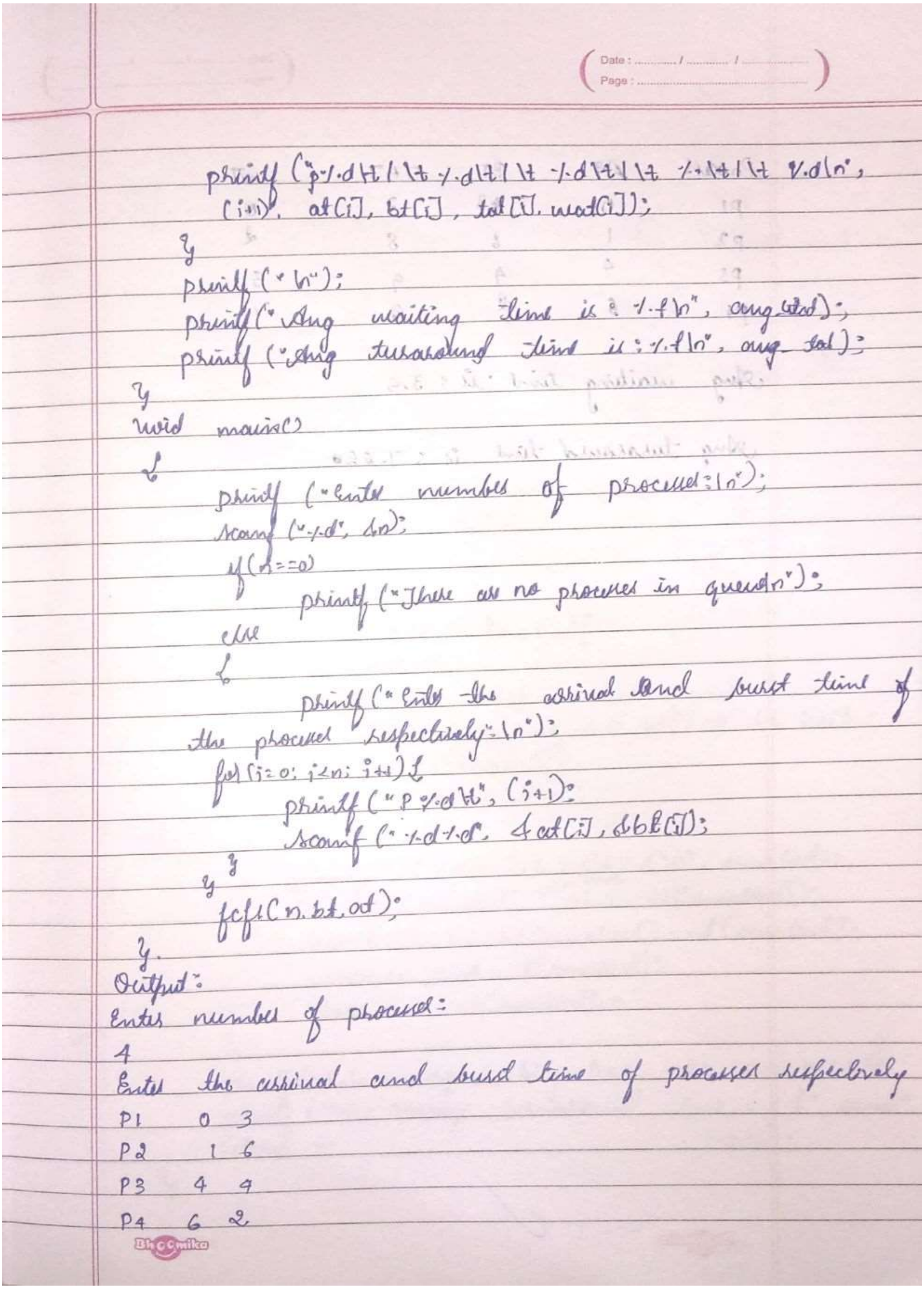


Date : ..../......../........

Page : ....

FCFS

```
#include<stdio.h>
#include<stdlib.h>
int cmpt[30], bt[30], at[30], tat[30], wat[30];
int n, i, sum
void fcfs(int n, int bt[n], int at[n])
{
        float avg_tat=0, avg_wat=0;
        for (i=0; i<n; i++)
        {
                sum+= bt[i];
                cmpt [i]= sum
        }
        for(i=0; i<n; i++)
        {
                tat [i] = cmpt[i]- at[i];
                wat[i] = tat[i]-bt[i];
        }
        printf("\n");
        for(i=0; i<n; i++)
        {
                avg_tat = avg_tat + tat[i];
                avg_wat = avg_wat + wat[i];
         }
        avg_tat = avg_tat/n;
        avg_wat = avg_wat/n;
        printf("PROCESS\t ARRIVAL TIME\t BURST TIME\t
        TURN AROUND TIME\t WAITING TIME\n");
        for(i=0; i<n; i++)
        {
```

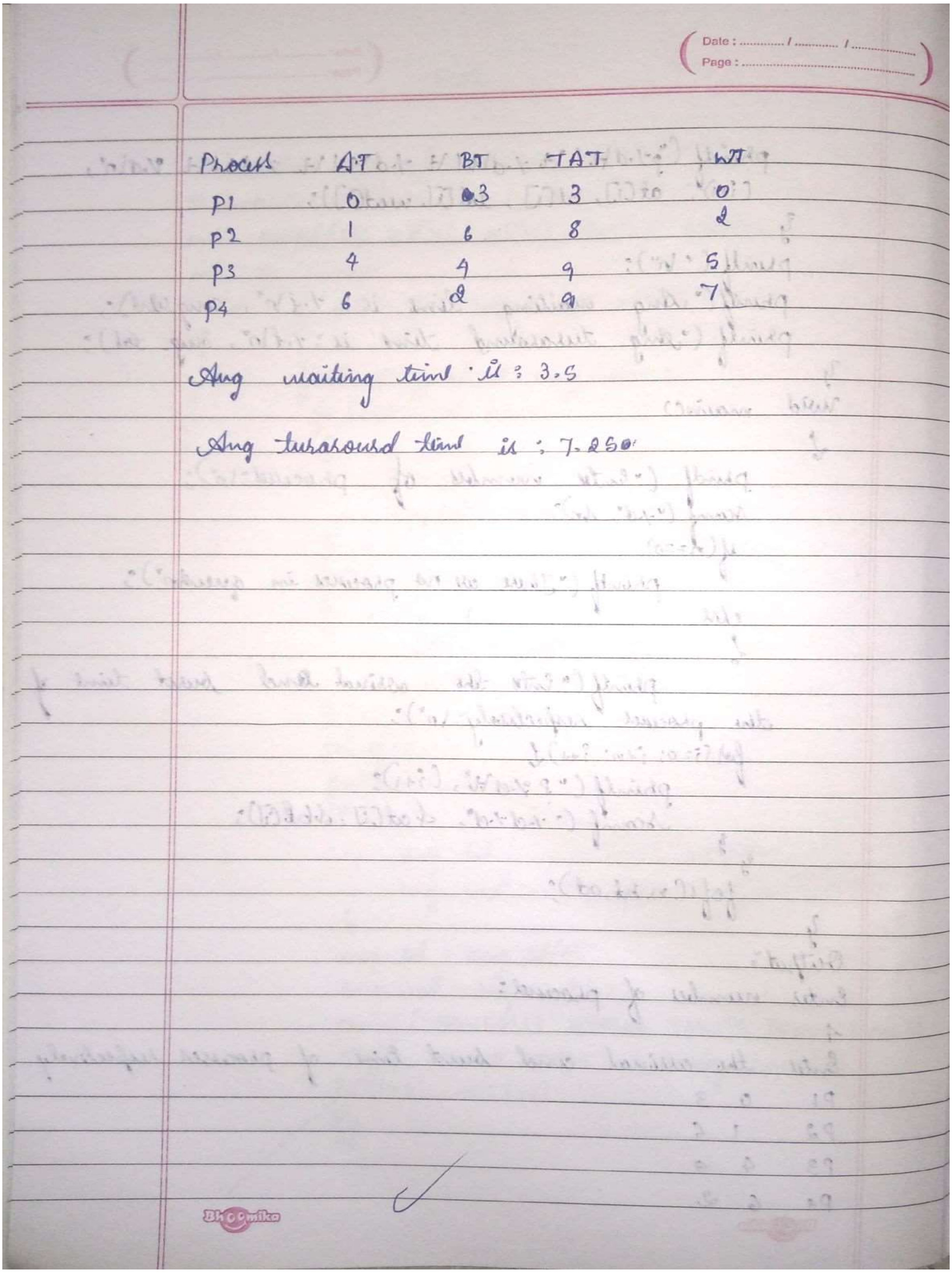


Date : ..../........./.........
Page : ....

```
    printf ("p%d\t\t %d\t\t %d\t\t %d\t\t %d\n",
     (i+1), at[i], bt[i], tat[i]. wat[i]]);
  }
  printf ("\n");
  printf("Avg waiting time is : %f\n", avg_wat);
  printf ("Avg turnaround time is : %f\n", avg_tat);
}
void main()
{
      printf ("Enter numbers of processes:\n");
      scanf ("%d", &n);
       if(n==0)
              printf ("There are no processes in queue\n");
      else
      {
              printf("Enter the arrival and burst time of
    the processes respectively:\n");
     for(i=0; i<n; i++){
           printf ("P %d\t", (i+1));
           scanf (" %d%d", &at[i], &bt[i]);
        }
    }
     fcfs(n,bt,at);
 }.
```

Output:

```
Enter number of processes:
4
Enter the arrival and burst time of processes respectively
P1    0  3
P2    1  6
P3    4  4
P4    6  2
```



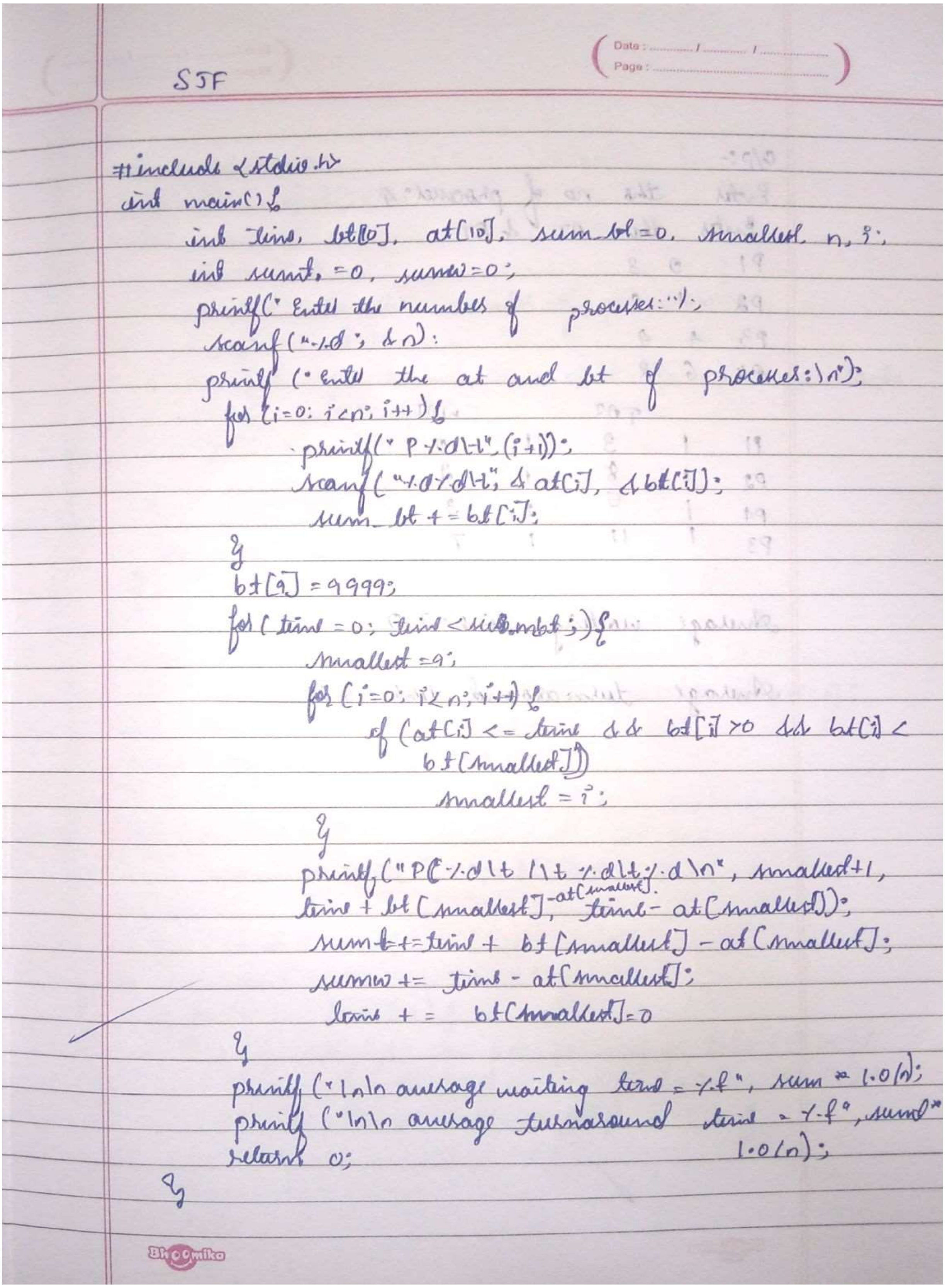
Date : ..../........../..........
Page : ....

Process	AT	BT	TAT	WT
P1	0	3	3	0
P2	1	6	8	2
P3	4	4	9	5
P4	6	2	9	7

Avg waiting time is : 3.5

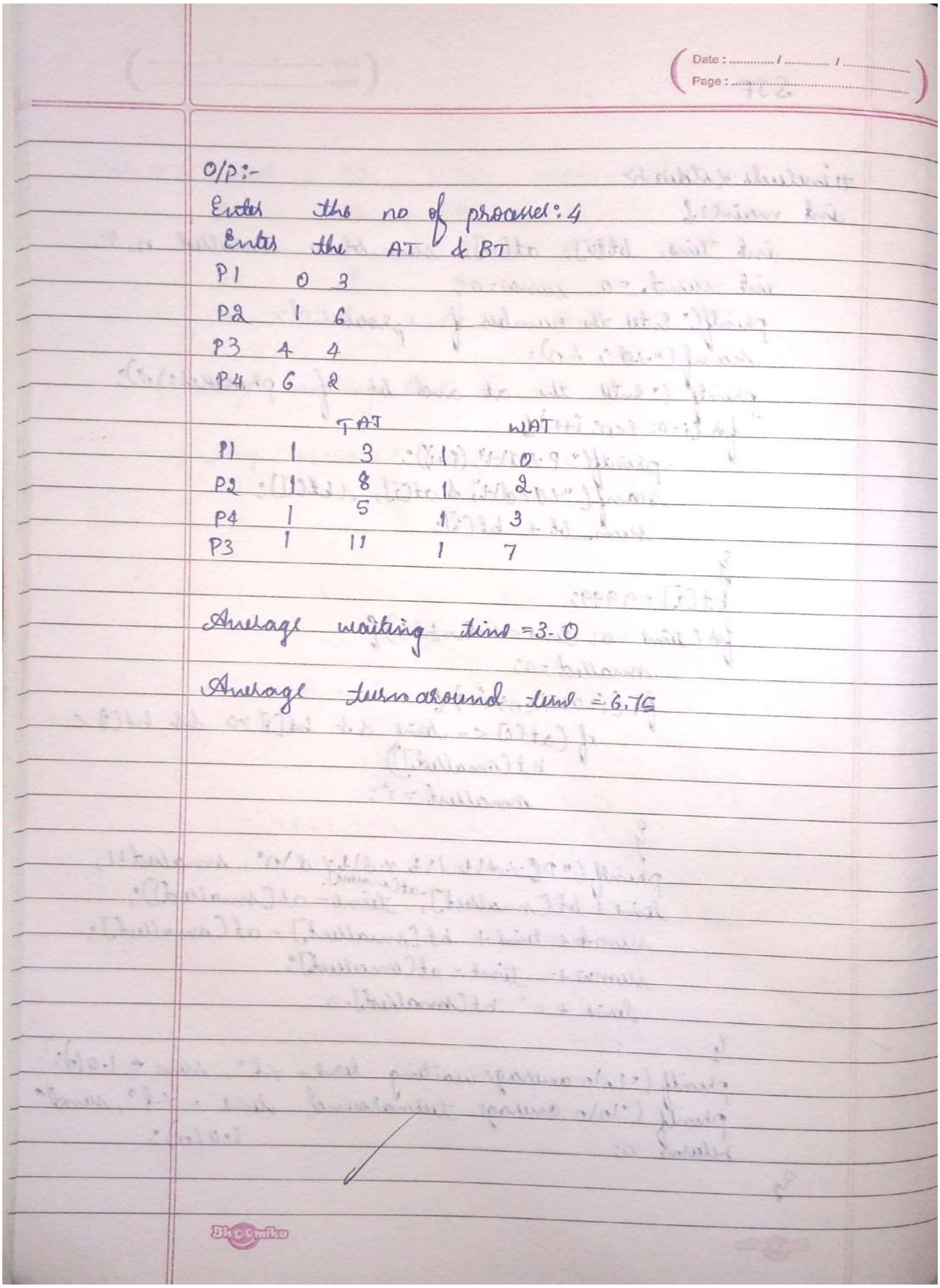
Avg turnaround time is : 7.250

✓



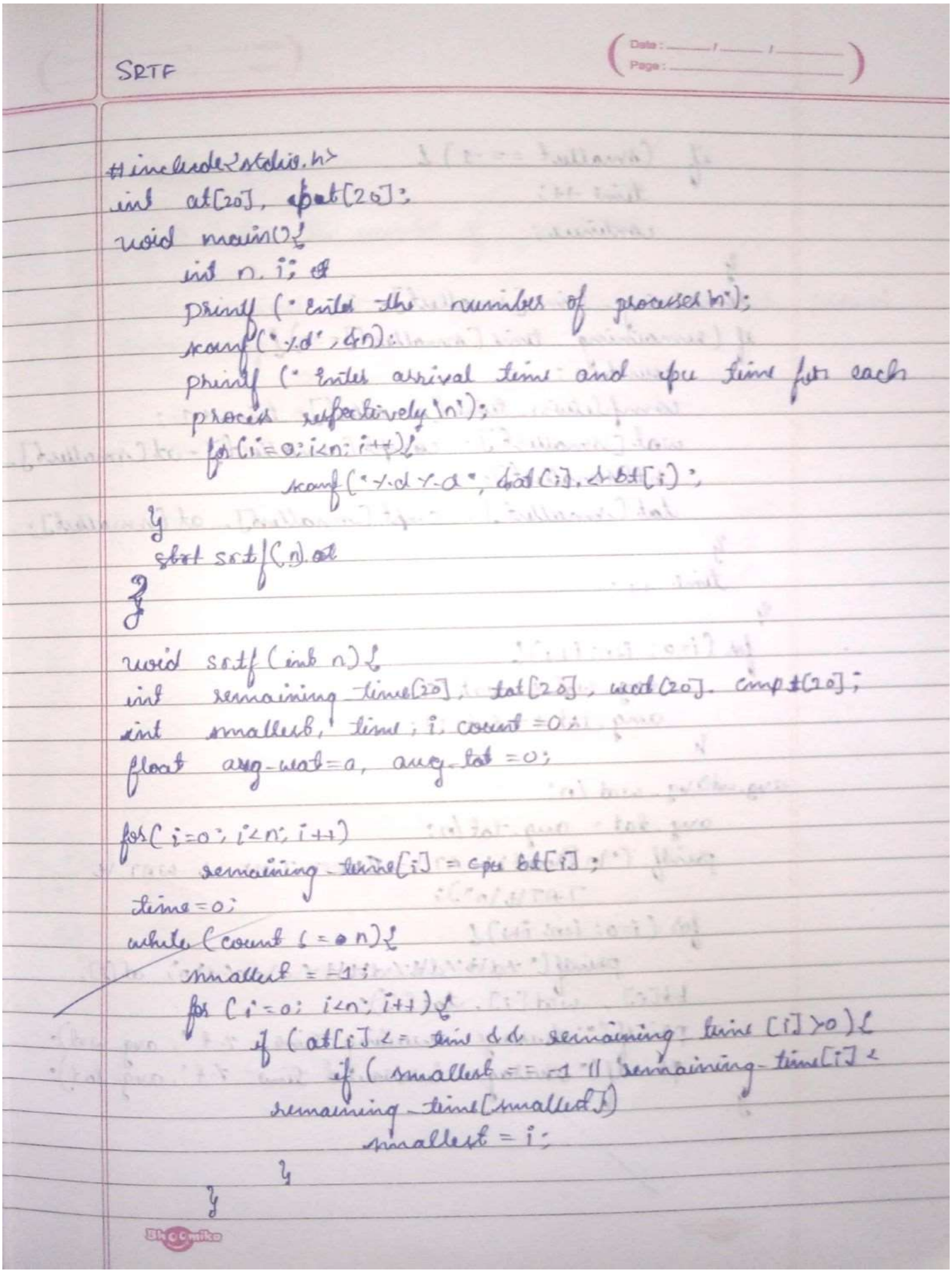
SJF

```
#include <stdio.h>
int main(){
        int time, bt[10], at[10], sum_bt=0, smallest, n, i;
        int sumt=0, sumw=0;
        printf(" Enter the numbers of processes:");
         scanf("%d", &n);
         printf(" Enter the at and bt of processes:\n");
          for(i=0; i<n; i++){
                printf(" P%d\t", (i+1));
                scanf("%d%d\t", &at[i], &bt[i]);
                 sum_bt += bt[i];
          }
          bt[9] = 9999;
          for( time = 0; time < sum_bt;){
                 smallest =9;
                 for (i=0; i<n; i++){
                       if (at[i] <= time && bt[i] >0 && bt[i] <
                            bt[smallest])
                              smallest = i;
                  }
                  printf("P[%d\t |\t %d\t%d\n", smallest+1,
                  time + bt[smallest] -at[smallest], time- at[smallest]);
                  sumt+=time + bt[smallest] - at[smallest];
                  sumw += time - at[smallest];
                   time += bt[smallest]=0
          }
          printf("\n\n average waiting time = %f", sum * 1.0/n);
          printf("\n\n average turnaround time = %f", sumt*
          return 0;                                   1.0/n);
}
```



Date : .... / .... / ....
Page : ....

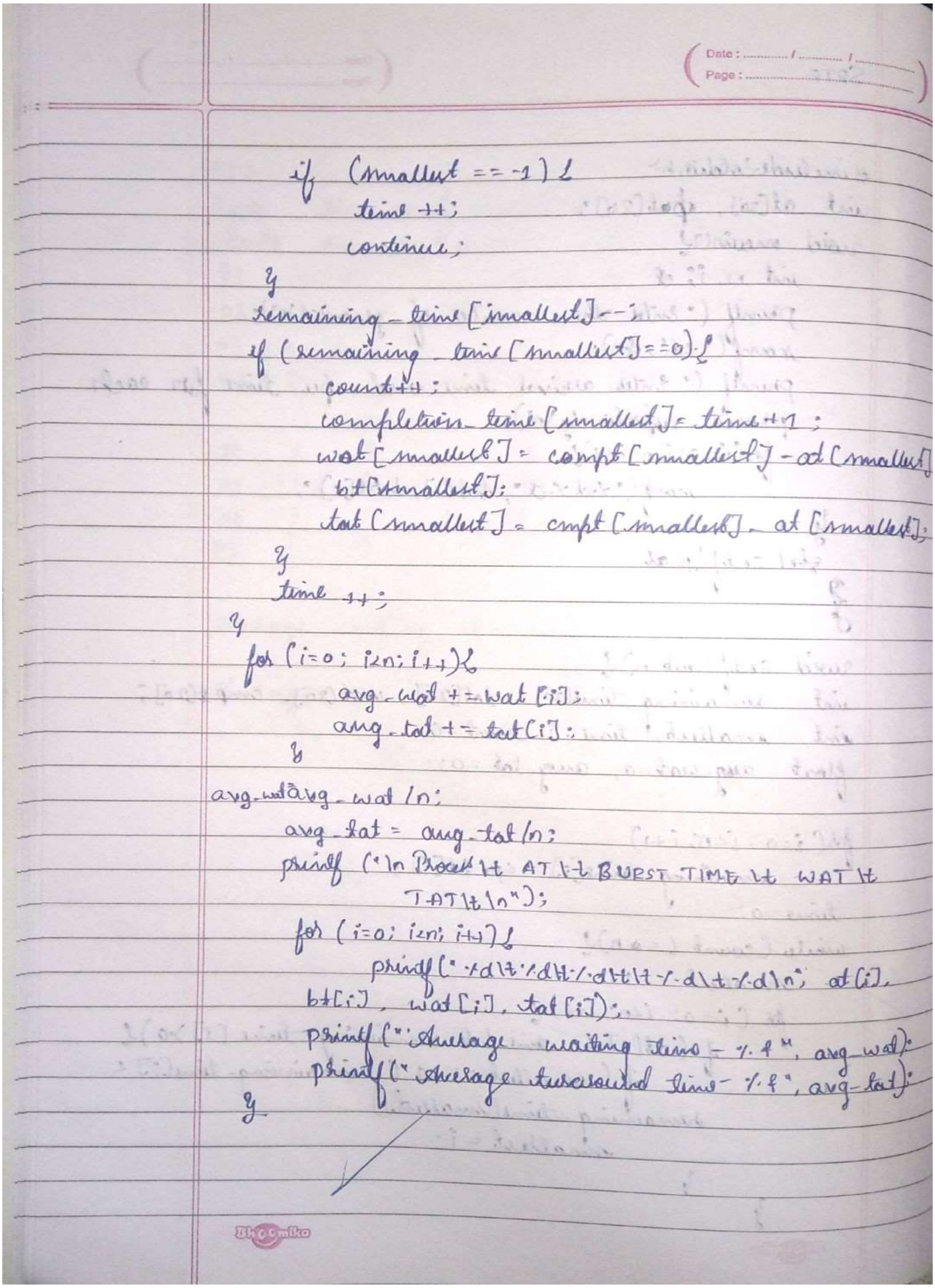
```
O/P:-
Enter the no of process: 4
Enter the AT & BT
P1   0  3
P2   1  6
P3   4  4
P4   6  2
            TAT          WAT
P1   1      3       1    0
P2   1      8       1    2
P4   1      5       1    3
P3   1      11      1    7

Average waiting time = 3.0

Average turnaround time = 6.75
```



SRTF

Date : ____/____/______
Page : ______________

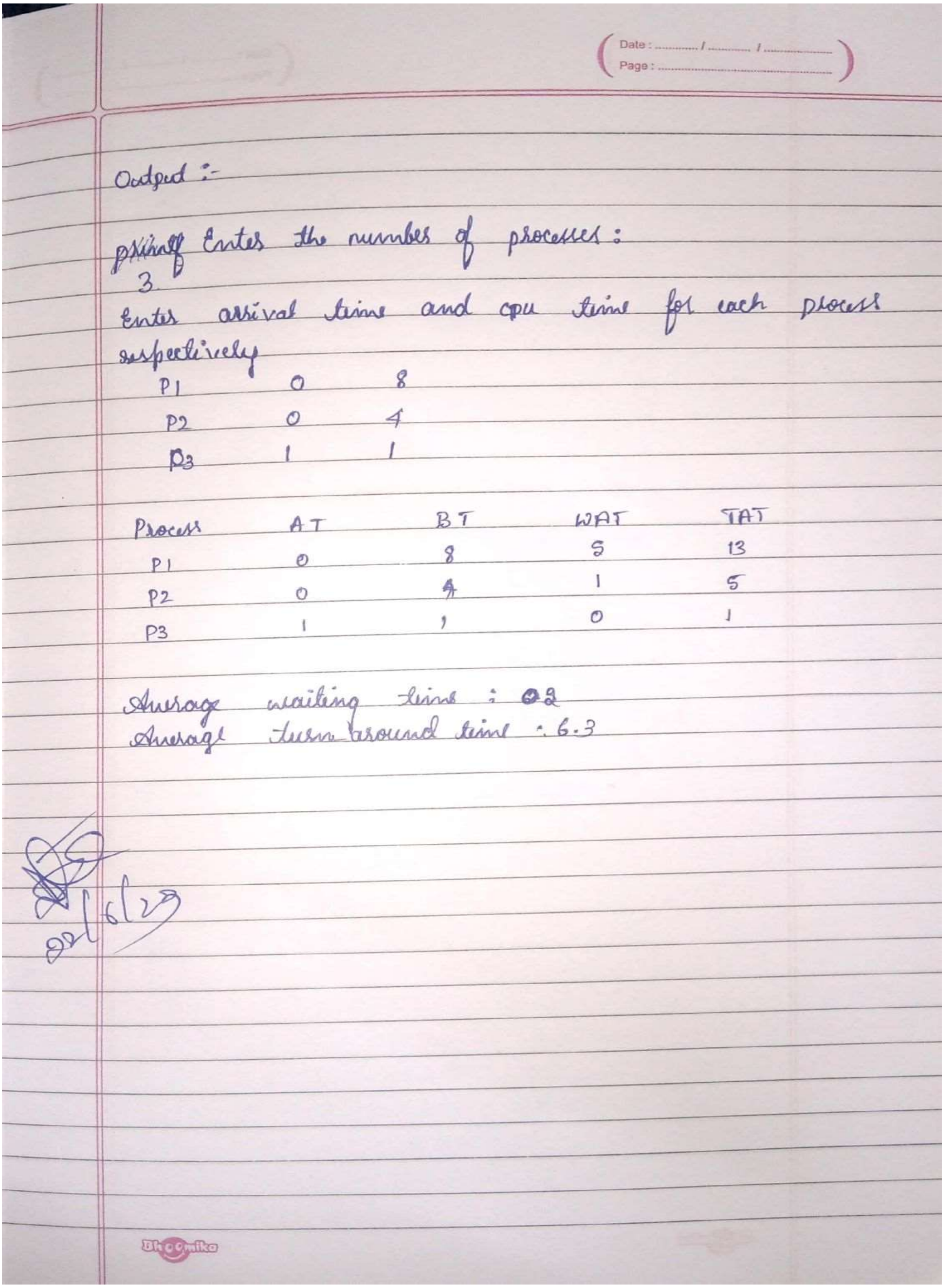
```
#include<stdio.h>
int at[20], cpbt[20];
void main(){
      int n, i; ~~t~~
      printf ("Enter the number of process\n");
      scanf("%d", &n);
      printf (" Enter arrival time and cpu time for each
      process respectively\n");
       for(i=0; i<n; i++){
              scanf("%d %d", &at[i], &bt[i]);
    }
    ~~sbrt~~ srtf(n).~~ot~~
}

void srtf(int n){
int   remaining_time[20], tat[20], wat[20], cmpt[20];
int   smallest, time; i, count =0;
float  avg_wat=0, avg_tat =0;

for( i=0; i<n; i++)
     remaining_time[i] = cpu bt[i];
time=0;
while (count != n){
     smallest = -1;
     for (i=0; i<n; i++){
          if (at[i] <= time && remaining_time [i] >0){
               if( smallest == -1 || remaining_time[i] <
          remaining_time[smallest])
                    smallest = i;
          }
     }
```



Date : ..../........../..........

Page : ....

```
    if (smallest == -1) {
        time ++;
        continue;
    }
    remaining_time [smallest]--;
    if (remaining_time [smallest]==0){
        count++;
        completion_time [smallest]= time+1 ;
        wat [smallest] = compt [smallest] - at [smallest]
         bt[smallest];
        tat [smallest] = cmpt [smallest] _ at [smallest];
    }
    time ++;
  }
  for (i=0; i<n; i++){
        avg_wat += wat [i];
        avg_tat += tat[i];
    }
avg_wat=avg_wat /n;
    avg_tat = avg_tat/n;
    printf ("\n Process\t AT\t BURST TIME \t WAT \t
                TAT\t \n");
    for (i=0; i<n; i++){
            printf(" %d\t %d\t %d\t\t %d\t %d\n", at[i],
    bt[i], wat[i], tat[i]);
    printf(" Average waiting time - %f", avg_wat);
    printf(" Average turnaround time- %f", avg_tat);
 }
```



Date : .... / .... / ....
Page : ....

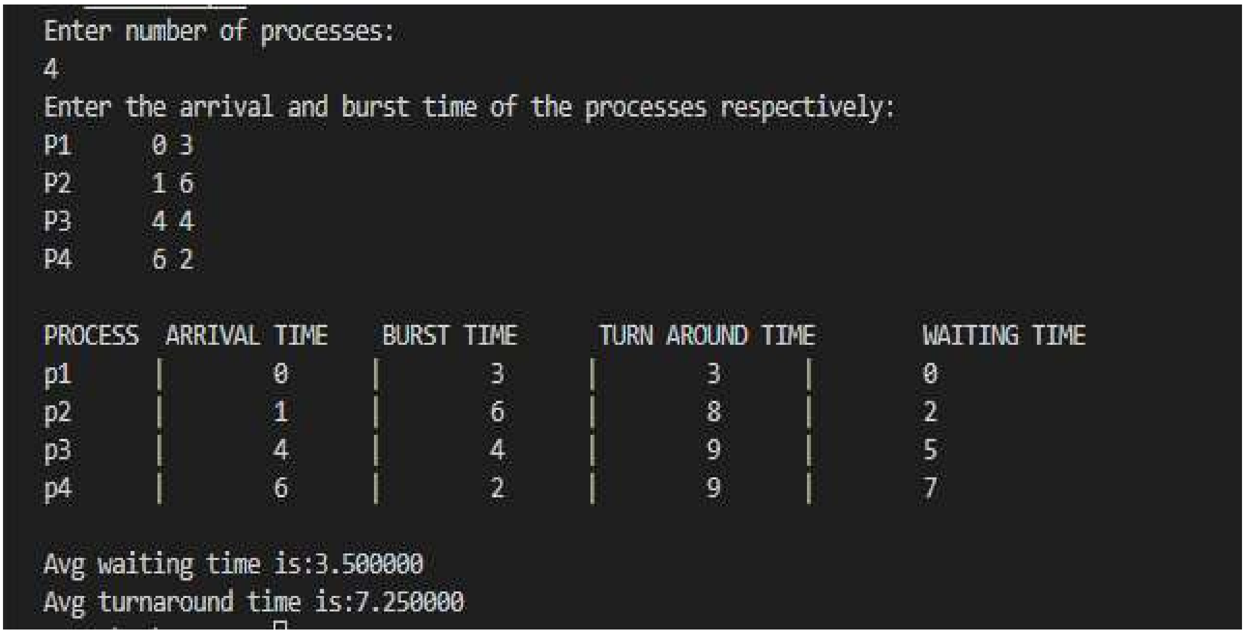
Output :-

```
~~printf~~ Enter the number of processes :
 3
Enter arrival time and cpu time for each process
respectively
   P1        0        8
   P2        0        4
   P3        1        1

 Process     AT        BT        WAT       TAT
  P1         0         8          5        13
  P2         0         4          1         5
  P3         1         1          0         1

Average waiting time : 2
Average turn around time : 6.3
```

22/6/23

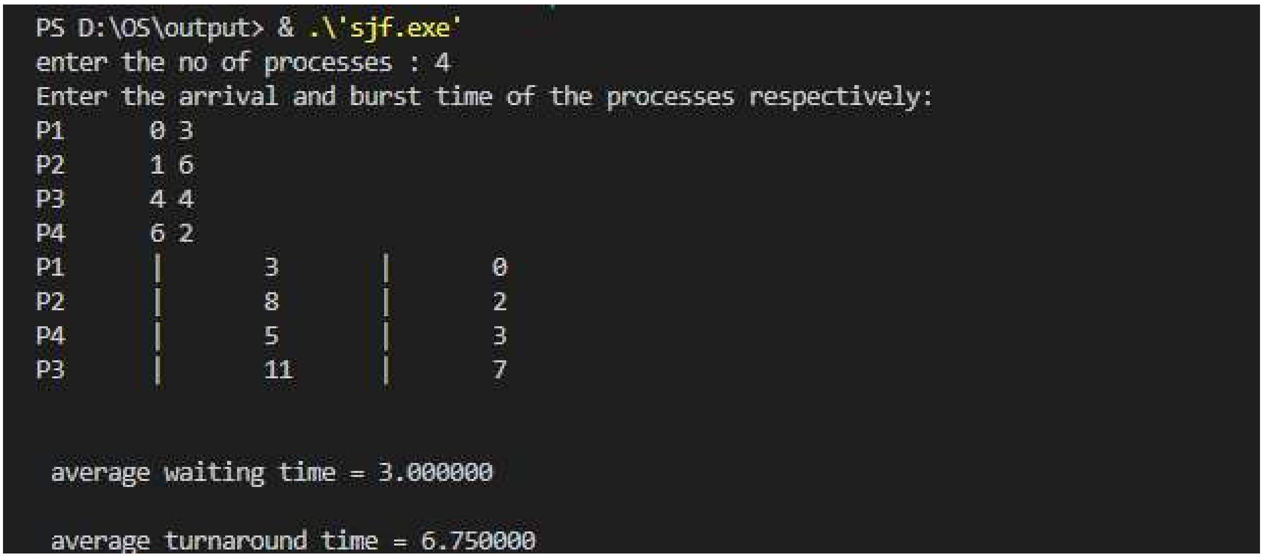
OUTPUT SCREENS

FCFS



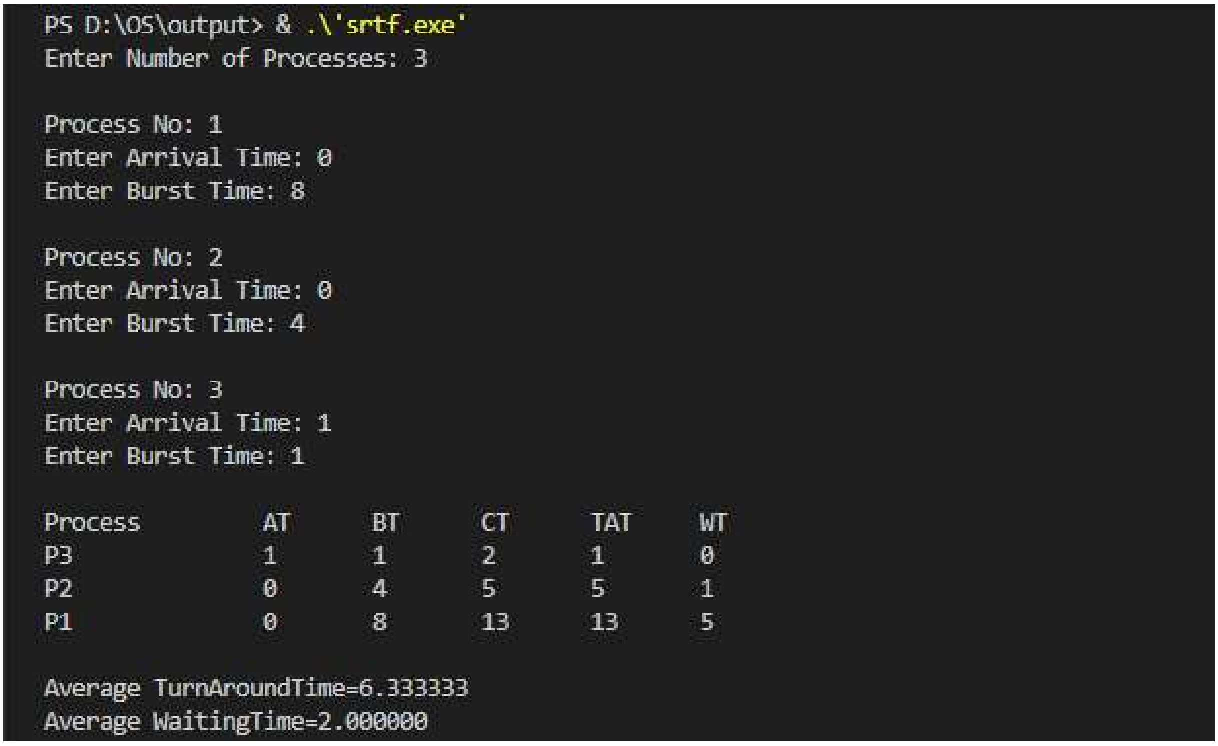
```
Enter number of processes:
4
Enter the arrival and burst time of the processes respectively:
P1     0 3
P2     1 6
P3     4 4
P4     6 2

PROCESS  ARRIVAL TIME    BURST TIME     TURN AROUND TIME       WAITING TIME
p1      |        0      |        3     |        3     |        0
p2      |        1      |        6     |        8     |        2
p3      |        4      |        4     |        9     |        5
p4      |        6      |        2     |        9     |        7

Avg waiting time is:3.500000
Avg turnaround time is:7.250000
```

SJF



```
PS D:\OS\output> & .\'sjf.exe'
enter the no of processes : 4
Enter the arrival and burst time of the processes respectively:
P1      0 3
P2      1 6
P3      4 4
P4      6 2
P1      |       3       |       0
P2      |       8       |       2
P4      |       5       |       3
P3      |       11      |       7


 average waiting time = 3.000000

 average turnaround time = 6.750000
```

SRTF



```
PS D:\OS\output> & .\'srtf.exe'
Enter Number of Processes: 3

Process No: 1
Enter Arrival Time: 0
Enter Burst Time: 8

Process No: 2
Enter Arrival Time: 0
Enter Burst Time: 4

Process No: 3
Enter Arrival Time: 1
Enter Burst Time: 1

Process         AT      BT      CT      TAT     WT
P3              1       1       2       1       0
P2              0       4       5       5       1
P1              0       8       13      13      5

Average TurnAroundTime=6.333333
Average WaitingTime=2.000000
```