Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai
Phule Pune University)

Operating System Lab

Assignment No. - 3

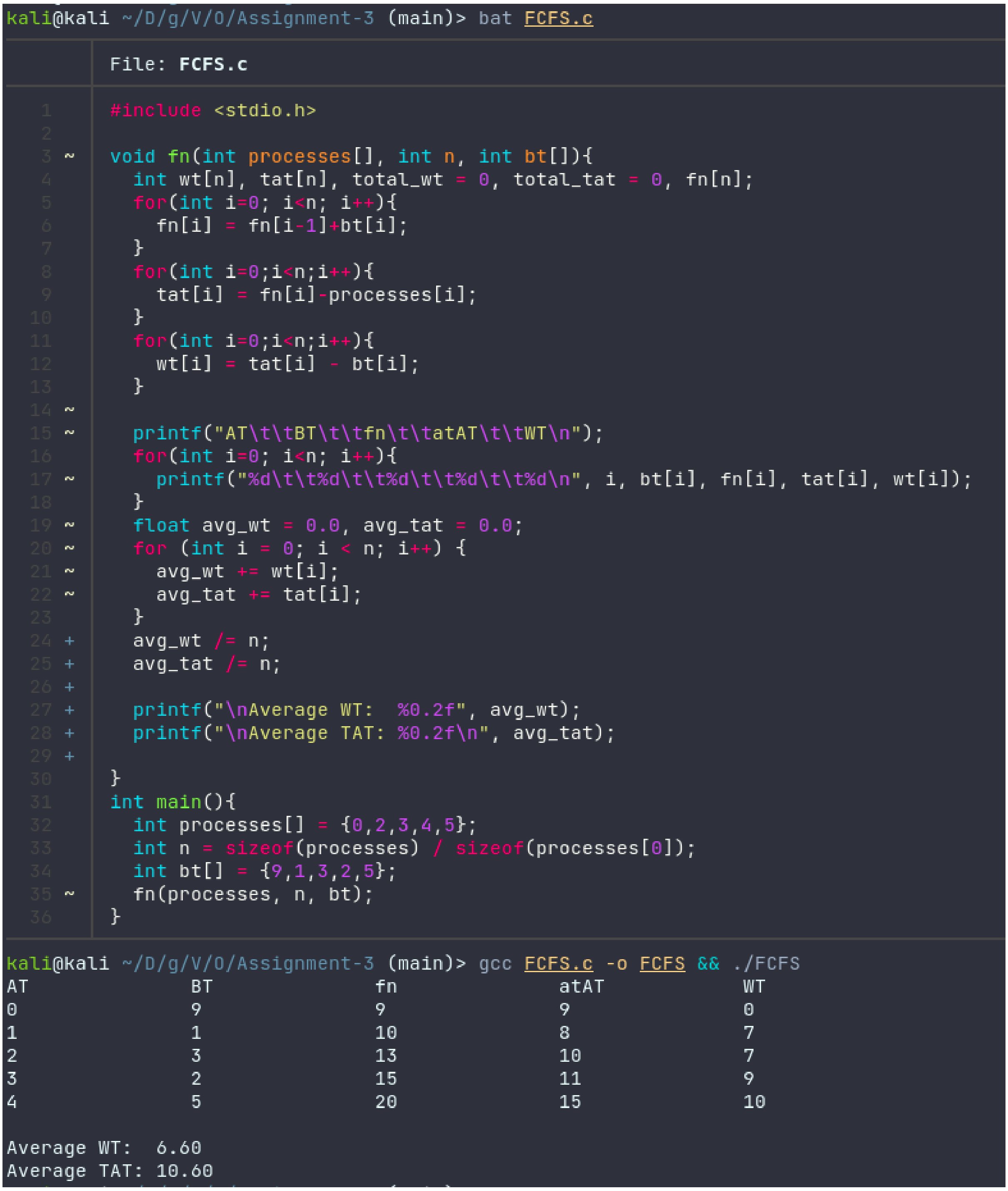
Name : Mayur Satish Khadde

Roll No : 21

Problem :

1. First Come First Serve Scheduling
2. Shortest Job First Preemptive Scheduling
3. Shortest Job First Non-Preemptive Scheduling
4. Round Robin Scheduling
5. Priority Based Scheduling – Non-Preemptive
6. Priority Based Scheduling - Preemptive

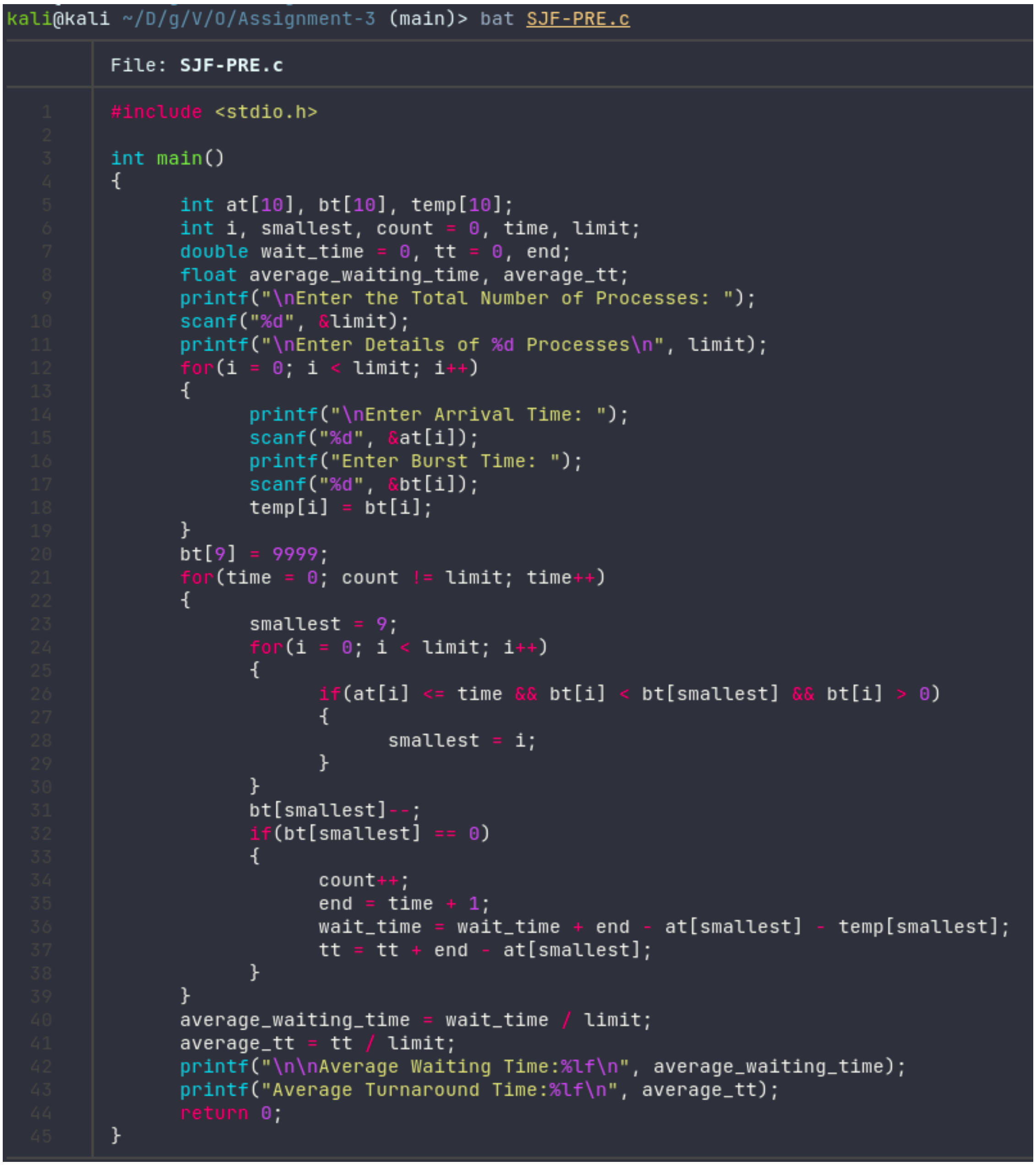
First Come First Serve Scheduling



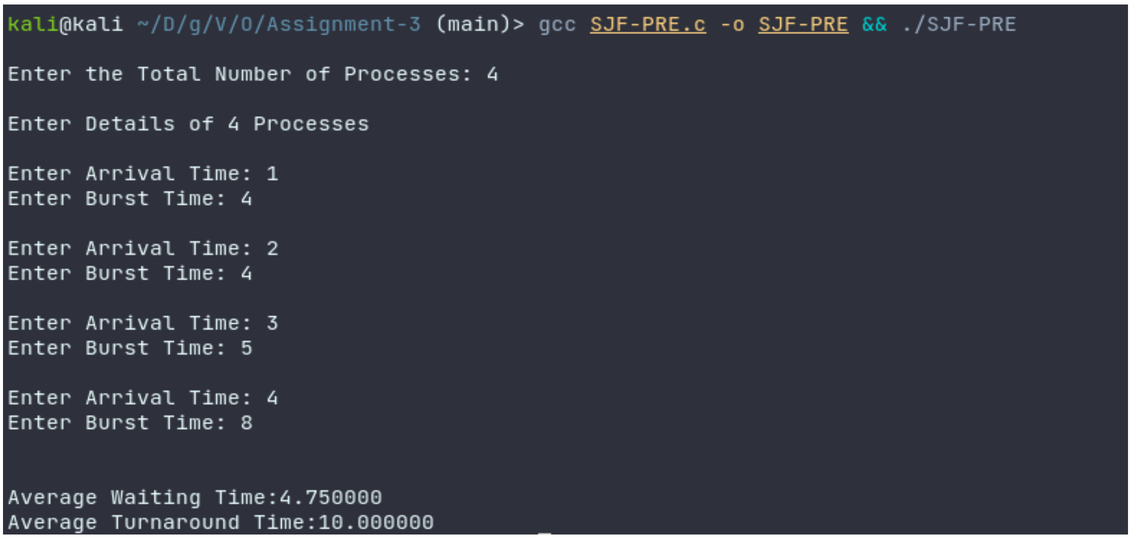
```
kali@kali ~/D/g/V/O/Assignment-3 (main)> bat FCFS.c
       File: FCFS.c

#include <stdio.h>

void fn(int processes[], int n, int bt[]){
  int wt[n], tat[n], total_wt = 0, total_tat = 0, fn[n];
  for(int i=0; i<n; i++){
    fn[i] = fn[i-1]+bt[i];
  }
  for(int i=0;i<n;i++){
    tat[i] = fn[i]-processes[i];
  }
  for(int i=0;i<n;i++){
    wt[i] = tat[i] - bt[i];
  }

  printf("AT\t\tBT\t\tfn\t\tatAT\t\tWT\n");
  for(int i=0; i<n; i++){
    printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i, bt[i], fn[i], tat[i], wt[i]);
  }
  float avg_wt = 0.0, avg_tat = 0.0;
  for (int i = 0; i < n; i++) {
    avg_wt += wt[i];
    avg_tat += tat[i];
  }
  avg_wt /= n;
  avg_tat /= n;

  printf("\nAverage WT:  %0.2f", avg_wt);
  printf("\nAverage TAT: %0.2f\n", avg_tat);

}
int main(){
  int processes[] = {0,2,3,4,5};
  int n = sizeof(processes) / sizeof(processes[0]);
  int bt[] = {9,1,3,2,5};
  fn(processes, n, bt);
}
```

```
kali@kali ~/D/g/V/O/Assignment-3 (main)> gcc FCFS.c -o FCFS && ./FCFS
AT              BT              fn              atAT            WT
0               9               9               9               0
1               1               10              8               7
2               3               13              10              7
3               2               15              11              9
4               5               20              15              10

Average WT:  6.60
Average TAT: 10.60
```

Shortest Job First Preemptive Scheduling

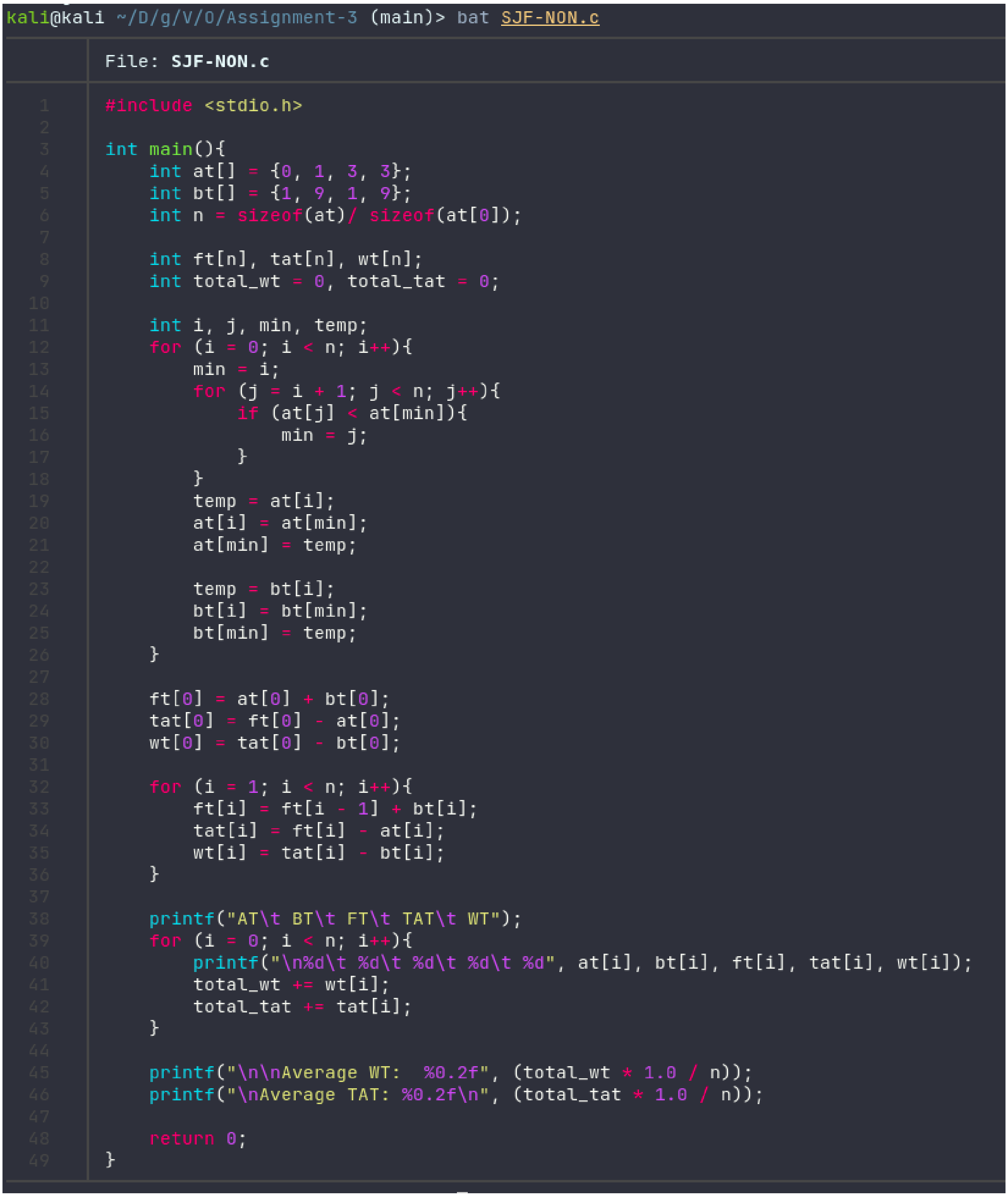


```
kali@kali ~/D/g/V/O/Assignment-3 (main)> bat SJF-PRE.c

File: SJF-PRE.c

#include <stdio.h>

int main()
{
      int at[10], bt[10], temp[10];
      int i, smallest, count = 0, time, limit;
      double wait_time = 0, tt = 0, end;
      float average_waiting_time, average_tt;
      printf("\nEnter the Total Number of Processes: ");
      scanf("%d", &limit);
      printf("\nEnter Details of %d Processes\n", limit);
      for(i = 0; i < limit; i++)
      {
            printf("\nEnter Arrival Time: ");
            scanf("%d", &at[i]);
            printf("Enter Burst Time: ");
            scanf("%d", &bt[i]);
            temp[i] = bt[i];
      }
      bt[9] = 9999;
      for(time = 0; count != limit; time++)
      {
            smallest = 9;
            for(i = 0; i < limit; i++)
            {
                  if(at[i] <= time && bt[i] < bt[smallest] && bt[i] > 0)
                  {
                        smallest = i;
                  }
            }
            bt[smallest]--;
            if(bt[smallest] == 0)
            {
                  count++;
                  end = time + 1;
                  wait_time = wait_time + end - at[smallest] - temp[smallest];
                  tt = tt + end - at[smallest];
            }
      }
      average_waiting_time = wait_time / limit;
      average_tt = tt / limit;
      printf("\n\nAverage Waiting Time:%lf\n", average_waiting_time);
      printf("Average Turnaround Time:%lf\n", average_tt);
      return 0;
}
```

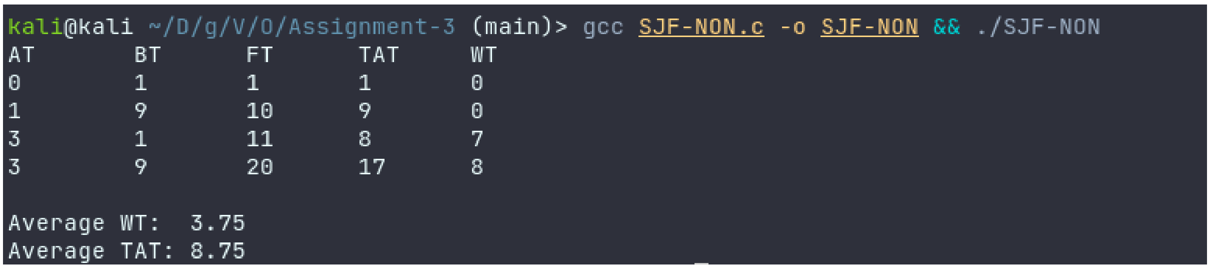


```
kali@kali ~/D/g/V/O/Assignment-3 (main)> gcc SJF-PRE.c -o SJF-PRE && ./SJF-PRE

Enter the Total Number of Processes: 4

Enter Details of 4 Processes

Enter Arrival Time: 1
Enter Burst Time: 4

Enter Arrival Time: 2
Enter Burst Time: 4

Enter Arrival Time: 3
Enter Burst Time: 5

Enter Arrival Time: 4
Enter Burst Time: 8


Average Waiting Time:4.750000
Average Turnaround Time:10.000000
```

Shortest Job First Non-Preemptive Scheduling

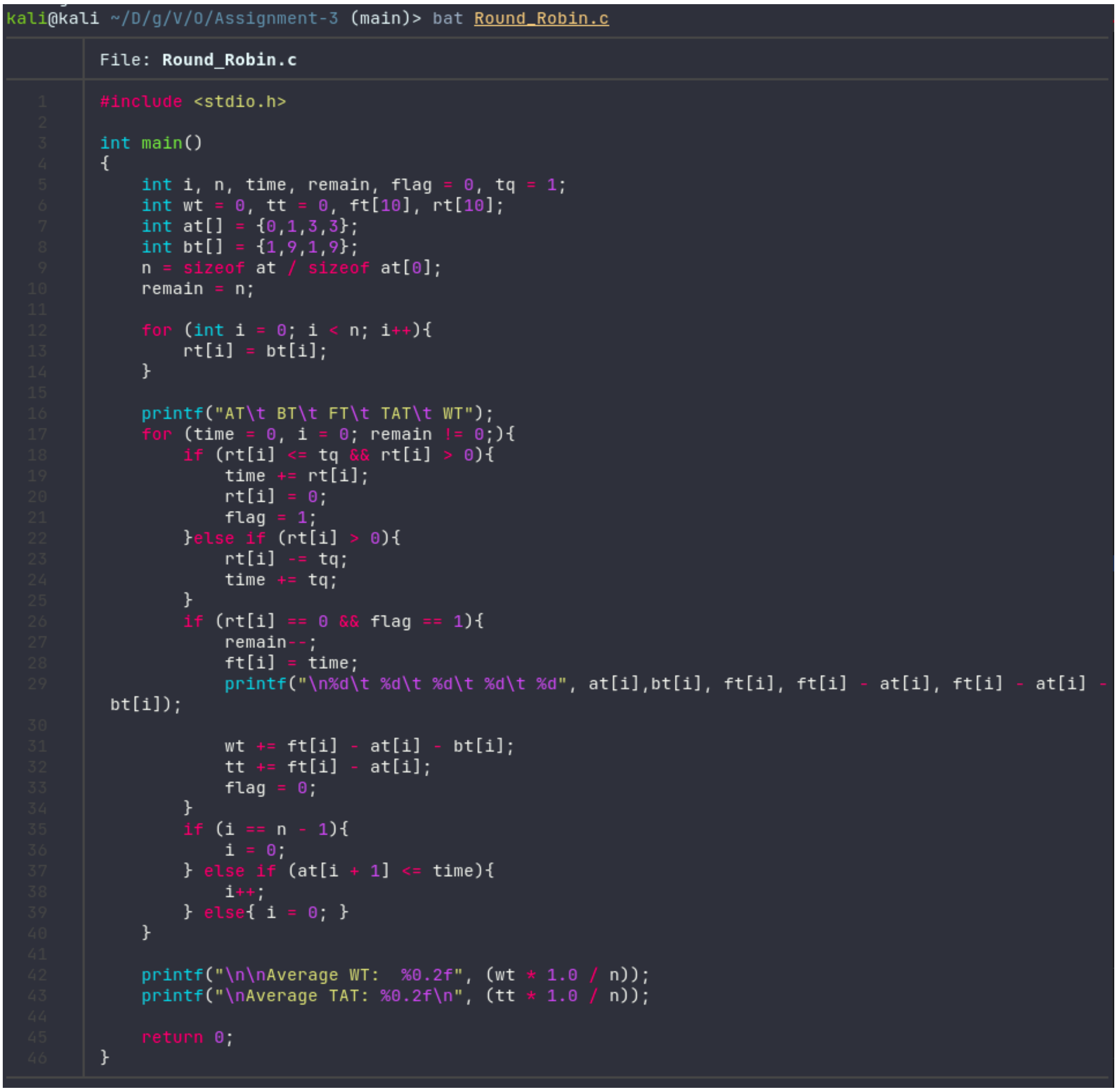


```
kali@kali ~/D/g/V/O/Assignment-3 (main)> bat SJF-NON.c

File: SJF-NON.c

#include <stdio.h>

int main(){
    int at[] = {0, 1, 3, 3};
    int bt[] = {1, 9, 1, 9};
    int n = sizeof(at)/ sizeof(at[0]);

    int ft[n], tat[n], wt[n];
    int total_wt = 0, total_tat = 0;

    int i, j, min, temp;
    for (i = 0; i < n; i++){
        min = i;
        for (j = i + 1; j < n; j++){
            if (at[j] < at[min]){
                min = j;
            }
        }
        temp = at[i];
        at[i] = at[min];
        at[min] = temp;

        temp = bt[i];
        bt[i] = bt[min];
        bt[min] = temp;
    }

    ft[0] = at[0] + bt[0];
    tat[0] = ft[0] - at[0];
    wt[0] = tat[0] - bt[0];

    for (i = 1; i < n; i++){
        ft[i] = ft[i - 1] + bt[i];
        tat[i] = ft[i] - at[i];
        wt[i] = tat[i] - bt[i];
    }

    printf("AT\t BT\t FT\t TAT\t WT");
    for (i = 0; i < n; i++){
        printf("\n%d\t %d\t %d\t %d\t %d", at[i], bt[i], ft[i], tat[i], wt[i]);
        total_wt += wt[i];
        total_tat += tat[i];
    }

    printf("\n\nAverage WT:  %0.2f", (total_wt * 1.0 / n));
    printf("\nAverage TAT: %0.2f\n", (total_tat * 1.0 / n));

    return 0;
}
```

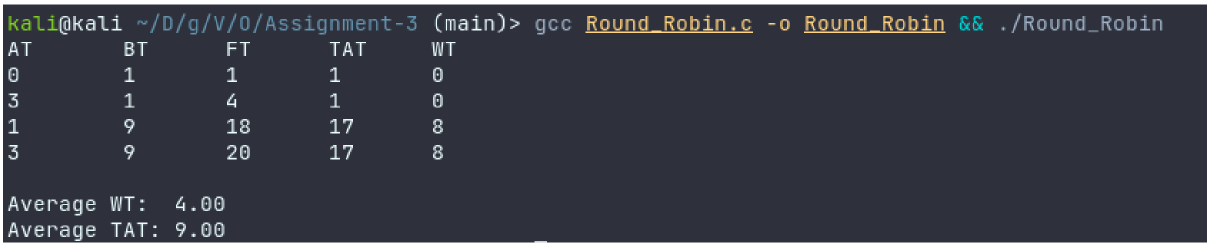


```
kali@kali ~/D/g/V/O/Assignment-3 (main)> gcc SJF-NON.c -o SJF-NON && ./SJF-NON
AT       BT      FT      TAT     WT
0        1       1       1       0
1        9       10      9       0
3        1       11      8       7
3        9       20      17      8

Average WT:  3.75
Average TAT: 8.75
```

Round Robin Scheduling

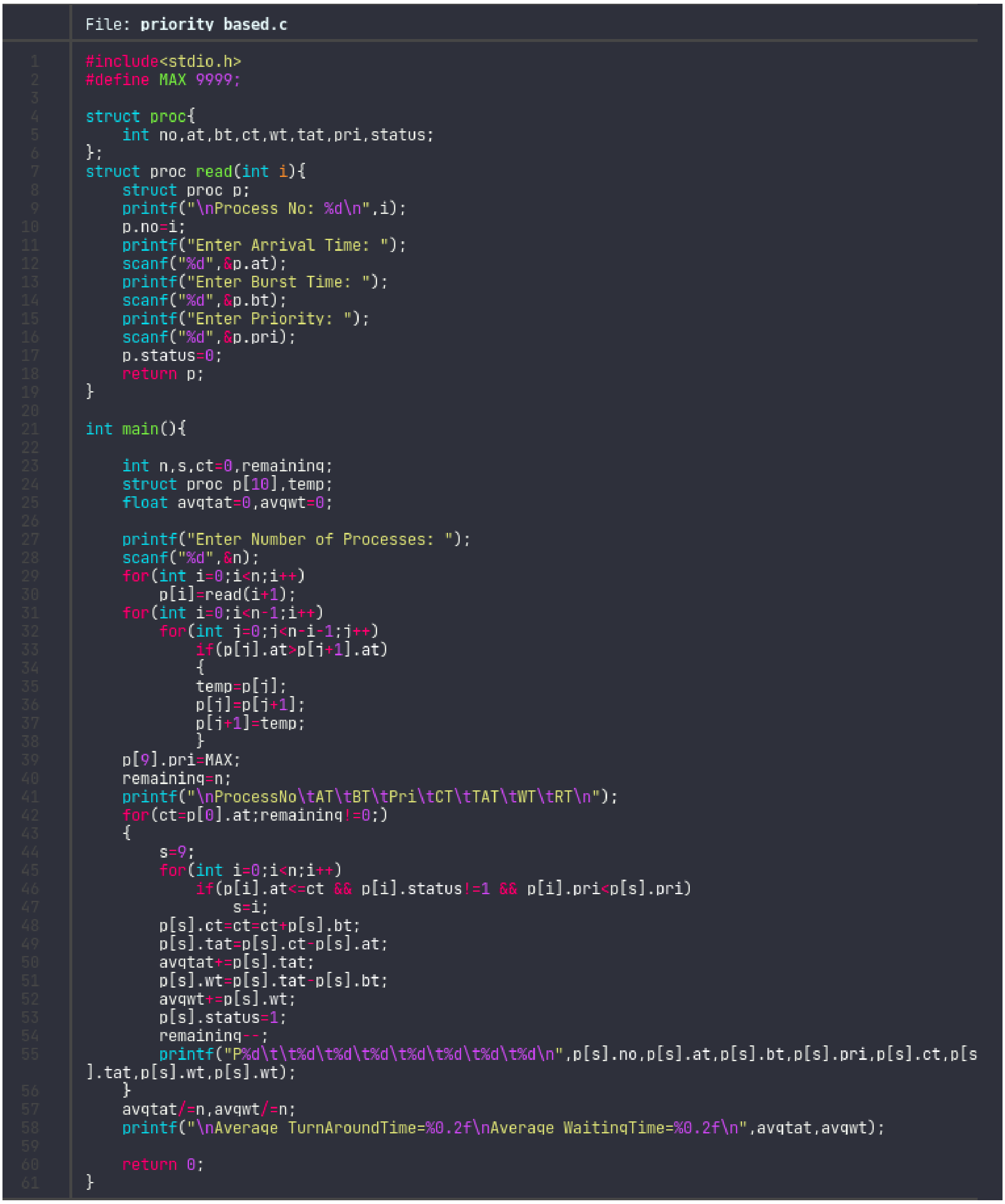


```
kali@kali ~/D/g/V/O/Assignment-3 (main)> bat Round_Robin.c

File: Round_Robin.c

#include <stdio.h>

int main()
{
    int i, n, time, remain, flag = 0, tq = 1;
    int wt = 0, tt = 0, ft[10], rt[10];
    int at[] = {0,1,3,3};
    int bt[] = {1,9,1,9};
    n = sizeof at / sizeof at[0];
    remain = n;

    for (int i = 0; i < n; i++){
        rt[i] = bt[i];
    }

    printf("AT\t BT\t FT\t TAT\t WT");
    for (time = 0, i = 0; remain != 0;){
        if (rt[i] <= tq && rt[i] > 0){
            time += rt[i];
            rt[i] = 0;
            flag = 1;
        }else if (rt[i] > 0){
            rt[i] -= tq;
            time += tq;
        }
        if (rt[i] == 0 && flag == 1){
            remain--;
            ft[i] = time;
            printf("\n%d\t %d\t %d\t %d\t %d", at[i],bt[i], ft[i], ft[i] - at[i], ft[i] - at[i] -
 bt[i]);

            wt += ft[i] - at[i] - bt[i];
            tt += ft[i] - at[i];
            flag = 0;
        }
        if (i == n - 1){
            i = 0;
        } else if (at[i + 1] <= time){
            i++;
        } else{ i = 0; }
    }

    printf("\n\nAverage WT:  %0.2f", (wt * 1.0 / n));
    printf("\nAverage TAT: %0.2f\n", (tt * 1.0 / n));

    return 0;
}
```



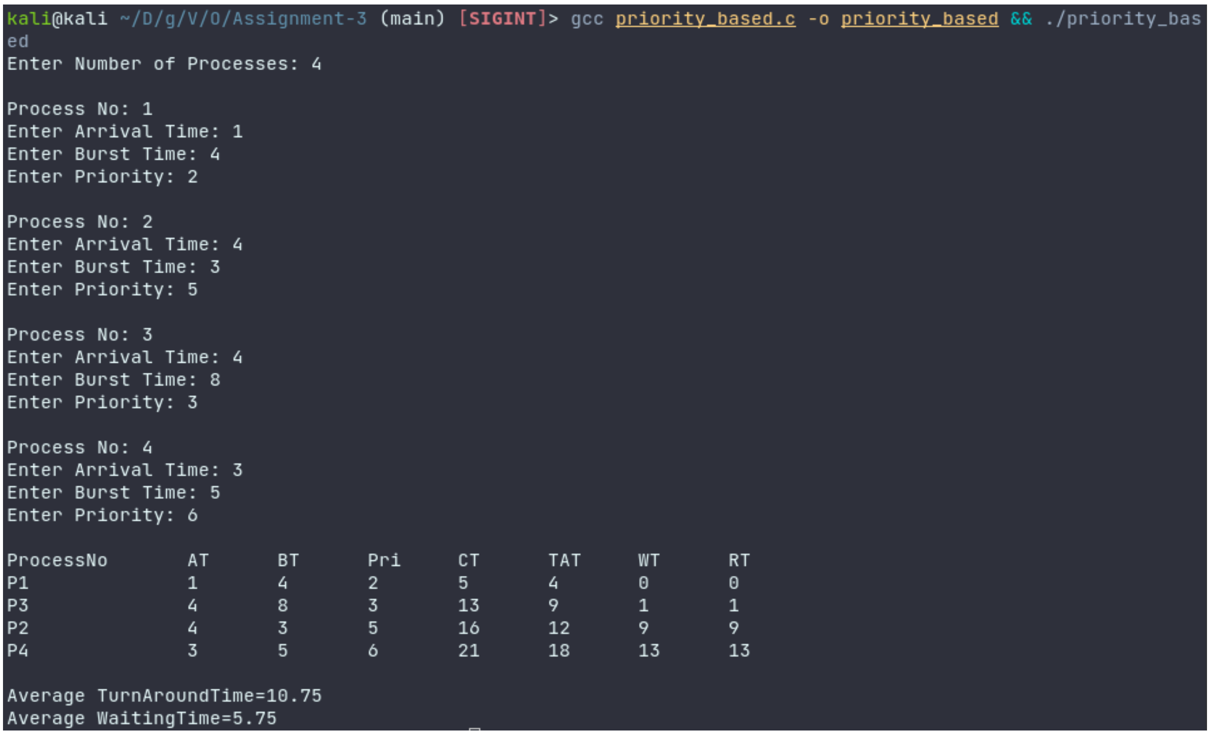
```
kali@kali ~/D/g/V/O/Assignment-3 (main)> gcc Round_Robin.c -o Round_Robin && ./Round_Robin
AT        BT        FT        TAT       WT
0         1         1         1         0
3         1         4         1         0
1         9         18        17        8
3         9         20        17        8

Average WT:  4.00
Average TAT: 9.00
```

Priority Based Scheduling – Non-Preemptive



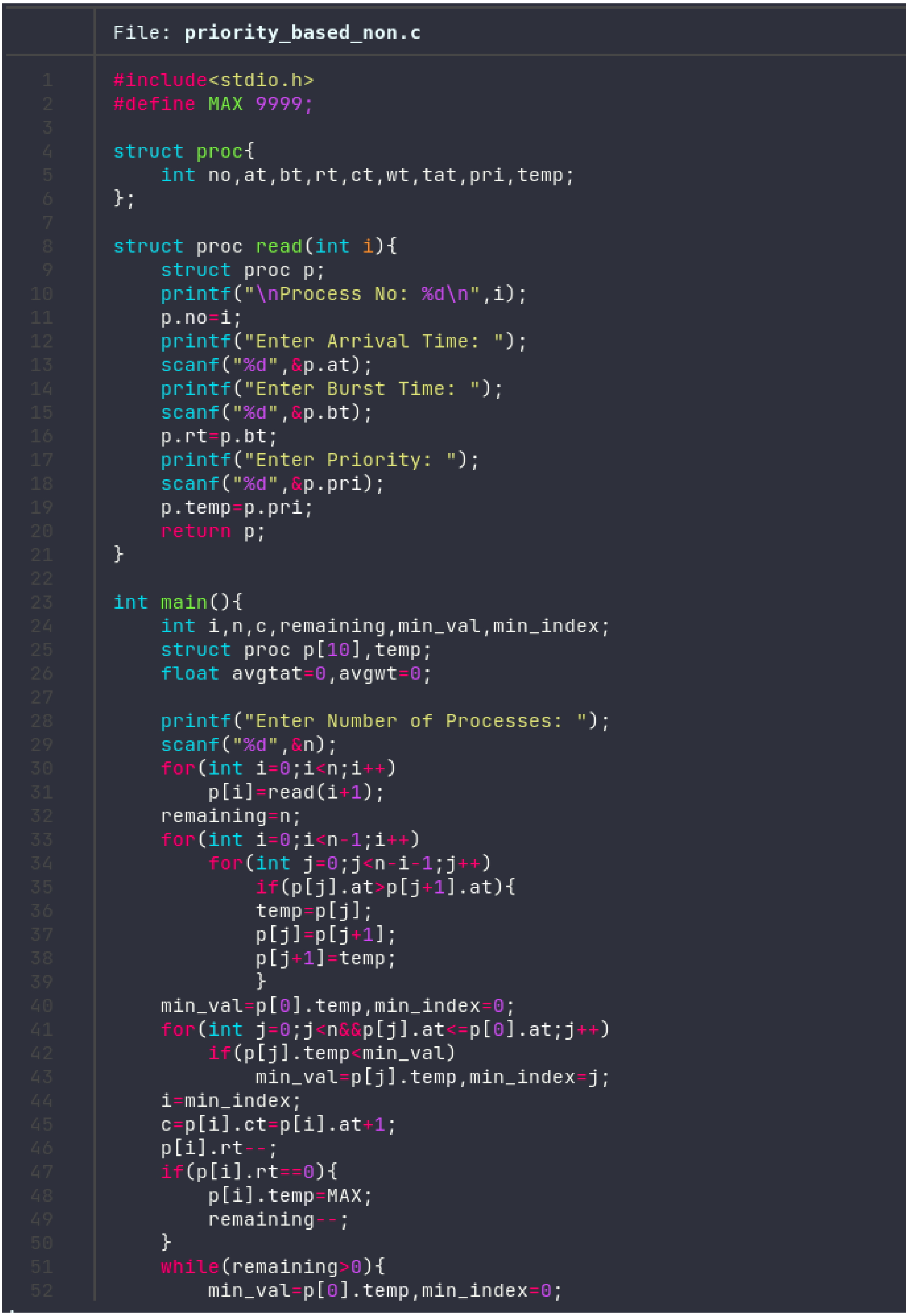
File: **priority based.c**

```c
#include<stdio.h>
#define MAX 9999;

struct proc{
    int no,at,bt,ct,wt,tat,pri,status;
};
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);
    printf("Enter Priority: ");
    scanf("%d",&p.pri);
    p.status=0;
    return p;
}

int main(){

    int n,s,ct=0,remaining;
    struct proc p[10],temp;
    float avgtat=0,avgwt=0;

    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;
            }
    p[9].pri=MAX;
    remaining=n;
    printf("\nProcessNo\tAT\tBT\tPri\tCT\tTAT\tWT\tRT\n");
    for(ct=p[0].at;remaining!=0;)
    {
        s=9;
        for(int i=0;i<n;i++)
            if(p[i].at<=ct && p[i].status!=1 && p[i].pri<p[s].pri)
                s=i;
        p[s].ct=ct=ct+p[s].bt;
        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;
        p[s].status=1;
        remaining--;
        printf("P%d\t\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",p[s].no,p[s].at,p[s].bt,p[s].pri,p[s].ct,p[s
].tat,p[s].wt,p[s].wt);
    }
    avgtat/=n,avgwt/=n;
    printf("\nAverage TurnAroundTime=%0.2f\nAverage WaitingTime=%0.2f\n",avgtat,avgwt);

    return 0;
}
```



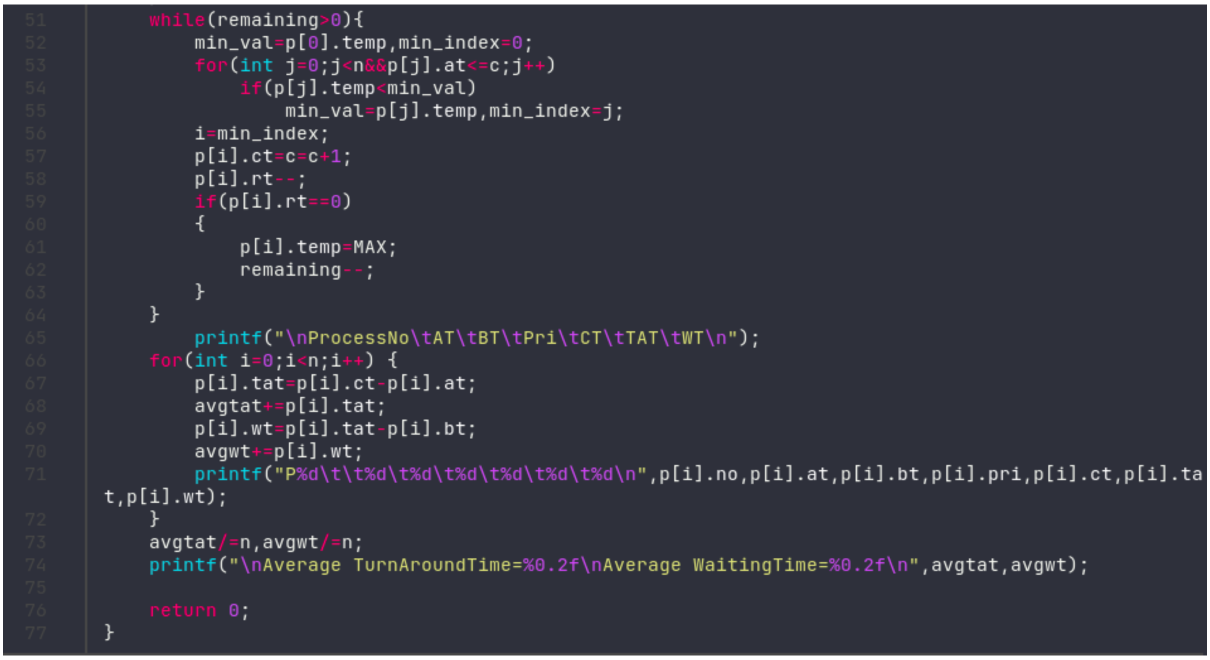
```
kali@kali ~/D/g/V/O/Assignment-3 (main) [SIGINT]> gcc priority_based.c -o priority_based && ./priority_bas
ed
Enter Number of Processes: 4

Process No: 1
Enter Arrival Time: 1
Enter Burst Time: 4
Enter Priority: 2

Process No: 2
Enter Arrival Time: 4
Enter Burst Time: 3
Enter Priority: 5

Process No: 3
Enter Arrival Time: 4
Enter Burst Time: 8
Enter Priority: 3

Process No: 4
Enter Arrival Time: 3
Enter Burst Time: 5
Enter Priority: 6

ProcessNo       AT      BT      Pri     CT      TAT     WT      RT
P1              1       4       2       5       4       0       0
P3              4       8       3       13      9       1       1
P2              4       3       5       16      12      9       9
P4              3       5       6       21      18      13      13

Average TurnAroundTime=10.75
Average WaitingTime=5.75
```

Priority Based Scheduling - Preemptive

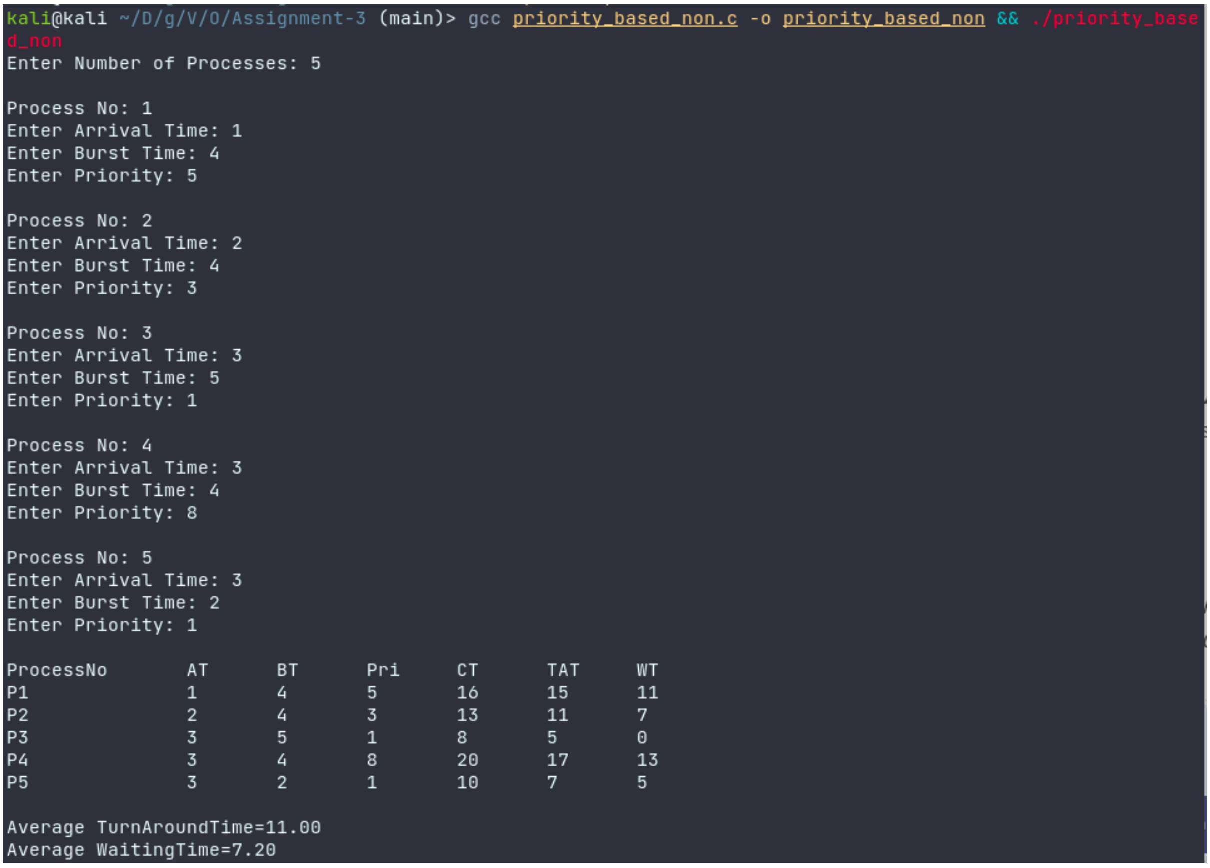


File: **priority_based_non.c**

```c
#include<stdio.h>
#define MAX 9999;

struct proc{
    int no,at,bt,rt,ct,wt,tat,pri,temp;
};

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;
    printf("Enter Priority: ");
    scanf("%d",&p.pri);
    p.temp=p.pri;
    return p;
}

int main(){
    int i,n,c,remaining,min_val,min_index;
    struct proc p[10],temp;
    float avgtat=0,avgwt=0;

    printf("Enter Number of Processes: ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        p[i]=read(i+1);
    remaining=n;
    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;
            }
    min_val=p[0].temp,min_index=0;
    for(int j=0;j<n&&p[j].at<=p[0].at;j++)
        if(p[j].temp<min_val)
            min_val=p[j].temp,min_index=j;
    i=min_index;
    c=p[i].ct=p[i].at+1;
    p[i].rt--;
    if(p[i].rt==0){
        p[i].temp=MAX;
        remaining--;
    }
    while(remaining>0){
        min_val=p[0].temp,min_index=0;
```



```c
    while(remaining>0){
        min_val=p[0].temp,min_index=0;
        for(int j=0;j<n&&p[j].at<=c;j++)
            if(p[j].temp<min_val)
                min_val=p[j].temp,min_index=j;
        i=min_index;
        p[i].ct=c=c+1;
        p[i].rt--;
        if(p[i].rt==0)
        {
            p[i].temp=MAX;
            remaining--;
        }
    }
        printf("\nProcessNo\tAT\tBT\tPri\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++) {
        p[i].tat=p[i].ct-p[i].at;
        avgtat+=p[i].tat;
        p[i].wt=p[i].tat-p[i].bt;
        avgwt+=p[i].wt;
        printf("P%d\t\t%d\t%d\t%d\t%d\t%d\t%d\n",p[i].no,p[i].at,p[i].bt,p[i].pri,p[i].ct,p[i].ta
t,p[i].wt);
    }
    avgtat/=n,avgwt/=n;
    printf("\nAverage TurnAroundTime=%0.2f\nAverage WaitingTime=%0.2f\n",avgtat,avgwt);

    return 0;
}
```



```
kali@kali ~/D/g/V/O/Assignment-3 (main)> gcc priority_based_non.c -o priority_based_non && ./priority_base
d_non
Enter Number of Processes: 5

Process No: 1
Enter Arrival Time: 1
Enter Burst Time: 4
Enter Priority: 5

Process No: 2
Enter Arrival Time: 2
Enter Burst Time: 4
Enter Priority: 3

Process No: 3
Enter Arrival Time: 3
Enter Burst Time: 5
Enter Priority: 1

Process No: 4
Enter Arrival Time: 3
Enter Burst Time: 4
Enter Priority: 8

Process No: 5
Enter Arrival Time: 3
Enter Burst Time: 2
Enter Priority: 1

ProcessNo       AT      BT      Pri     CT      TAT     WT
P1              1       4       5       16      15      11
P2              2       4       3       13      11      7
P3              3       5       1       8       5       0
P4              3       4       8       20      17      13
P5              3       2       1       10      7       5

Average TurnAroundTime=11.00
Average WaitingTime=7.20
```