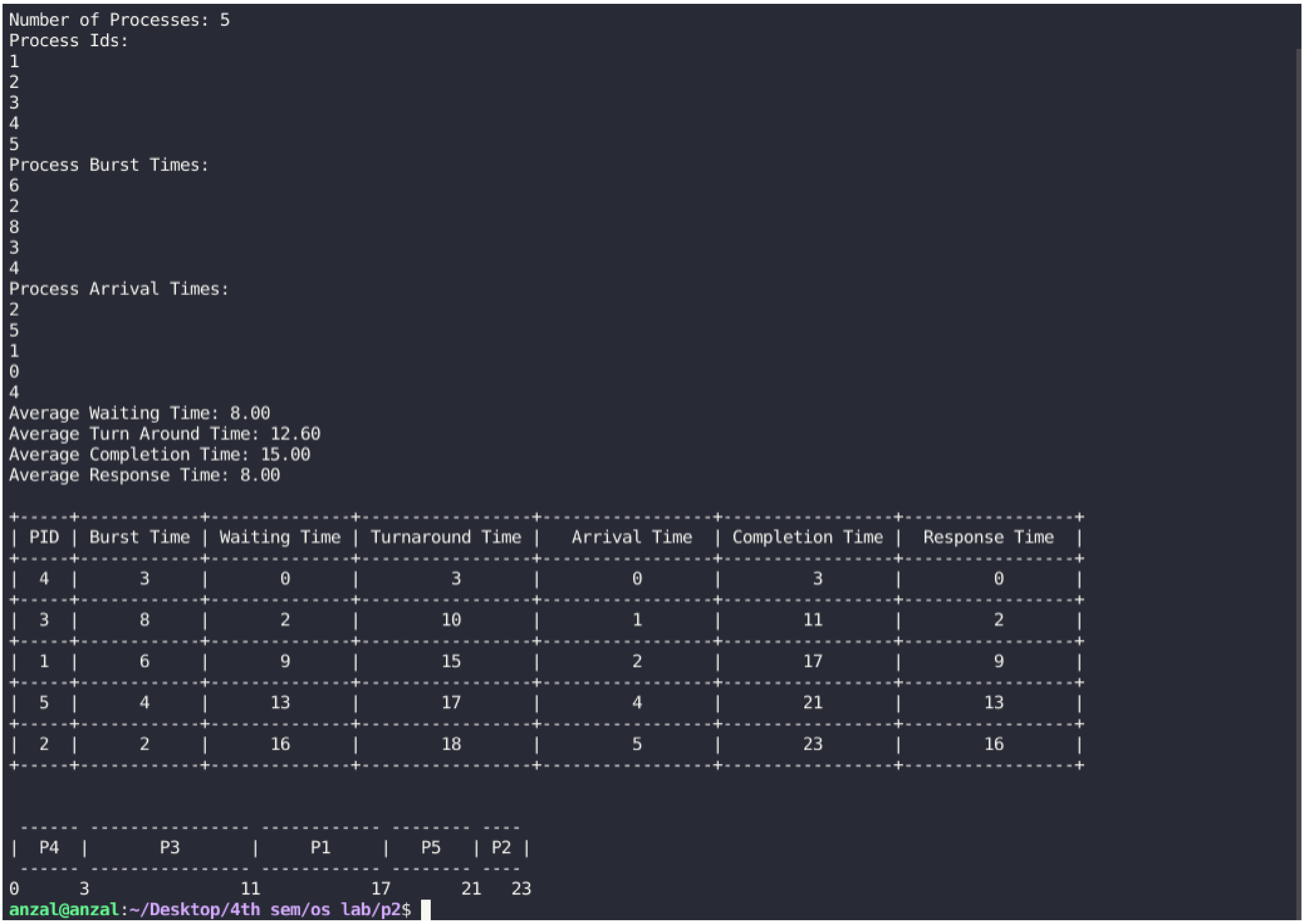
20BCS009 ANZAL HUSAIN ABIDI

```
#include <bits/stdc++.h>
using namespace std;

const int N=100005;
//  Write a program to implement the First Come First Serve
scheduling algorithm and find the average
// turnaround time, waiting time, completion time and response
time for overall process. Also Print Gantt
// chart for it.
int n;
struct process
{
    int id;
    int burst_time;
    int arrival_time;
    int waiting_time;
    int finishing_time;
    int turn_around_time;
    int completion_time;
};
process P[N];

void sorter( process P[],int n)
{
    int i, j,temp;
for(i = 0; i<n; i++) {
   for(j = i+1; j<n; j++)
   {
      if(P[j].arrival_time < P[i].arrival_time) {
         temp = P[i].arrival_time;
      P[i].arrival_time = P[j].arrival_time;
      P[j].arrival_time = temp;
      //
      temp = P[i].id;
      P[i].id = P[j].id;
      P[j].id = temp;
    }
  }
}

}

void FCFS()
{
   double total_waiting_time = 0.0;
   double total_turn_around_time = 0.0;
   double total_response_time = 0.0;
   double total_completion_time = 0.0;
   sorter(P,n);

   for(int i=0; i<n; i++)
   {
      P[i].finishing_time = P[i-1].finishing_time +
P[i].burst_time;
      P[i].turn_around_time = P[i].finishing_time -
P[i].arrival_time;
      P[i].waiting_time = P[i].turn_around_time - P[i].burst_time;
      if(P[i].waiting_time<0){P[i].waiting_time=0;}
      if(i==0){P[i].completion_time=P[i].burst_time;}
      else{P[i].completion_time=P[i].burst_time+P[i-
1].completion_time;}
      total_waiting_time += P[i].waiting_time;
      total_turn_around_time += P[i].turn_around_time;
```

```
        total_completion_time+=P[i].completion_time;

    }
    cout<<fixed<<setprecision(2);
    cout<<"Average Waiting Time: "<<(total_waiting_time/n)<<"\
n";
    cout<<"Average Turn Around Time:
"<<(total_turn_around_time/n)<<"\n";
    cout<<"Average Completion Time:
"<<(total_completion_time/n)<<"\n";
    cout<<"Average Response Time:
"<<(total_waiting_time/n)<<"\n";
    cout<<"\n";

    return;
}



void print_table(process P[], int n)
{
    int i;

    puts("+-----+------------+--------------+-----------------
+-----------------+-----------------+-----------------+");
    puts("| PID | Burst Time | Waiting Time | Turnaround Time |
Arrival Time  | Completion Time |  Response Time  |");
    puts("+-----+------------+--------------+-----------------
+-----------------+-----------------+-----------------+");

    for(i=0; i<n; i++) {
        printf("| %2d  |     %2d     |      %2d      |        %2d        |
%2d       |        %2d       |        %2d       |\n"
```

```
            , P[i].id, P[i].burst_time, P[i].waiting_time,
P[i].turn_around_time,  P[i].arrival_time, P[i].completion_time,
P[i].waiting_time );
        puts("+-----+------------+--------------+-----------------
+-----------------+-----------------+-----------------+");
    }
    cout<<"\n";
}

void print_gantt_chart(process P[], int n)
{
    cout<<"\n";
    int i, j;

    printf(" ");
    for(i=0; i<n; i++) {
        for(j=0; j<P[i].burst_time; j++) printf("--");
        printf(" ");
    }
    printf("\n|");



    for(i=0; i<n; i++) {
        for(j=0; j<P[i].burst_time - 1; j++) printf(" ");
        printf("P%d", P[i].id);
        for(j=0; j<P[i].burst_time - 1; j++) printf(" ");
        printf("|");
    }
    printf("\n ");

    for(i=0; i<n; i++) {
        for(j=0; j<P[i].burst_time; j++) printf("--");
        printf(" ");
    }
    printf("\n");



    printf("0");
    for(i=0; i<n; i++) {
        for(j=0; j<P[i].burst_time; j++) printf("  ");
        if(P[i].completion_time > 9) printf("\b");
        printf("%d", P[i].completion_time);

    }
    printf("\n");

}



int main()
{
    cout<<"Number of Processes: ";
    cin>>n;

    cout<<"Process Ids:\n";
    for(int i=0; i<n; i++) cin>>P[i].id;

    cout<<"Process Burst Times:\n";
    for(int i=0; i<n; i++) cin>>P[i].burst_time;

    cout<<"Process Arrival Times:\n";
    for(int i=0; i<n; i++) cin>>P[i].arrival_time;

    FCFS();
    print_table(P,n);
    print_gantt_chart(P,n);
    return 0;
}
```



```
Number of Processes: 5
Process Ids:
1
2
3
4
5
Process Burst Times:
6
2
8
3
4
Process Arrival Times:
2
5
1
0
4
Average Waiting Time: 8.00
Average Turn Around Time: 12.60
Average Completion Time: 15.00
Average Response Time: 8.00

+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+
| PID | Burst Time | Waiting Time | Turnaround Time |   Arrival Time  | Completion Time |  Response Time  |
+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+
|  4  |      3     |       0      |         3       |         0       |         3       |         0       |
+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+
|  3  |      8     |       2      |        10       |         1       |        11       |         2       |
+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+
|  1  |      6     |       9      |        15       |         2       |        17       |         9       |
+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+
|  5  |      4     |       13     |        17       |         4       |        21       |        13       |
+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+
|  2  |      2     |       16     |        18       |         5       |        23       |        16       |
+-----+------------+--------------+-----------------+-----------------+-----------------+-----------------+



 ------ ---------------- ------------ -------- ----
|  P4  |       P3       |     P1     |   P5   | P2 |
 ------ ---------------- ------------ -------- ----
0      3               11           17       21   23
anzal@anzal:~/Desktop/4th sem/os lab/p2$
```