

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

int n;
struct process
{
    int id;
    int burst_time;
    int arrival_time;
    int waiting_time;
    int turn_around_time;
    int completion_time;
};
process P[50];

bool btimeSort(process a, process b){
    return a.burst_time < b.burst_time;
}
bool atimeSort(process a, process b){
    return a.arrival_time < b.arrival_time;
}
bool pidSort(process a, process b){
    return a.id < b.id;
}
bool ctSort(process a, process b){
    return a.completion_time < b.completion_time;
}
void SJF(int n)
{
    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;
    sort(P, P+n, btimeSort);
    sort(P, P+n, atimeSort);

    int ttime=0, i;
    int j, tArray[n];
    for(i=0; i<n; i++){
        j=i;
        while(P[j].arrival_time <= ttime && j!=n){
            j++;
        }
        sort(P+i, P+j, btimeSort);
        tArray[i]=ttime;
```

```

    }
    sort(P+i,P+j,btimeSort);
    tArray[i]=ttime;
    ttime+=P[i].burst_time;
}
tArray[i] = ttime;

for(i=0; i<n; i++)
{
    P[i].completion_time=tArray[i+1];
    total_completion_time+=tArray[i+1];
    P[i].turn_around_time=tArray[i]-P[i].arrival_time+P[i].burst_time;
    total_turn_around_time+=tArray[i]-P[i].arrival_time+P[i].burst_time;
    P[i].waiting_time=tArray[i]-P[i].arrival_time;
    total_waiting_time+=tArray[i]-P[i].arrival_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;
    sort(P,P+n,pidSort);
    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 |",
            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)
{
    sort(P,P+n,ctSort);
    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 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;  
  
    SJF(n);  
    print_table(P,n);  
    print_gantt_chart(P,n);  
  
    return 0;  
}
```

```
anzal@anzal:~/4th sem/os lab/p3$ cd "/home/anzal/Desktop/4th sem/os lab/p3/" && g++ sjf_nonprim.cpp -o sjf_nonprim && "/home/anzal/Desktop/4th sem/os lab/p3/"sjf_nonprim
```

Number of Processes: 5

Process Ids:

1  
2  
3  
4  
5

Process Burst Times:

5  
2  
4  
1  
7

Process Arrival Times:

0  
1  
2  
3  
4

Average Waiting Time: 4.20

Average Turn Around Time: 8.00

Average Completion Time: 10.00

Average Response Time: 4.20

PID	Burst Time	Waiting Time	Turnaround Time	Arrival Time	Completion Time	Response Time
1	5	0	5	0	5	0
2	2	5	7	1	8	5
3	4	6	10	2	12	6
4	1	2	3	3	6	2
5	7	8	15	4	19	8

P1	P2	P3	P5
5	6	8	12
0			19