Sidharth

2K18/MC/114

Experiment 5

Aim: Implement FCFS and Non preemptive Shortest Job first algorithm.

Input: Sequence of Processes with their arrival time and burst time

Output: Sequence of processes executing Average Waiting time and Average Turnaround time

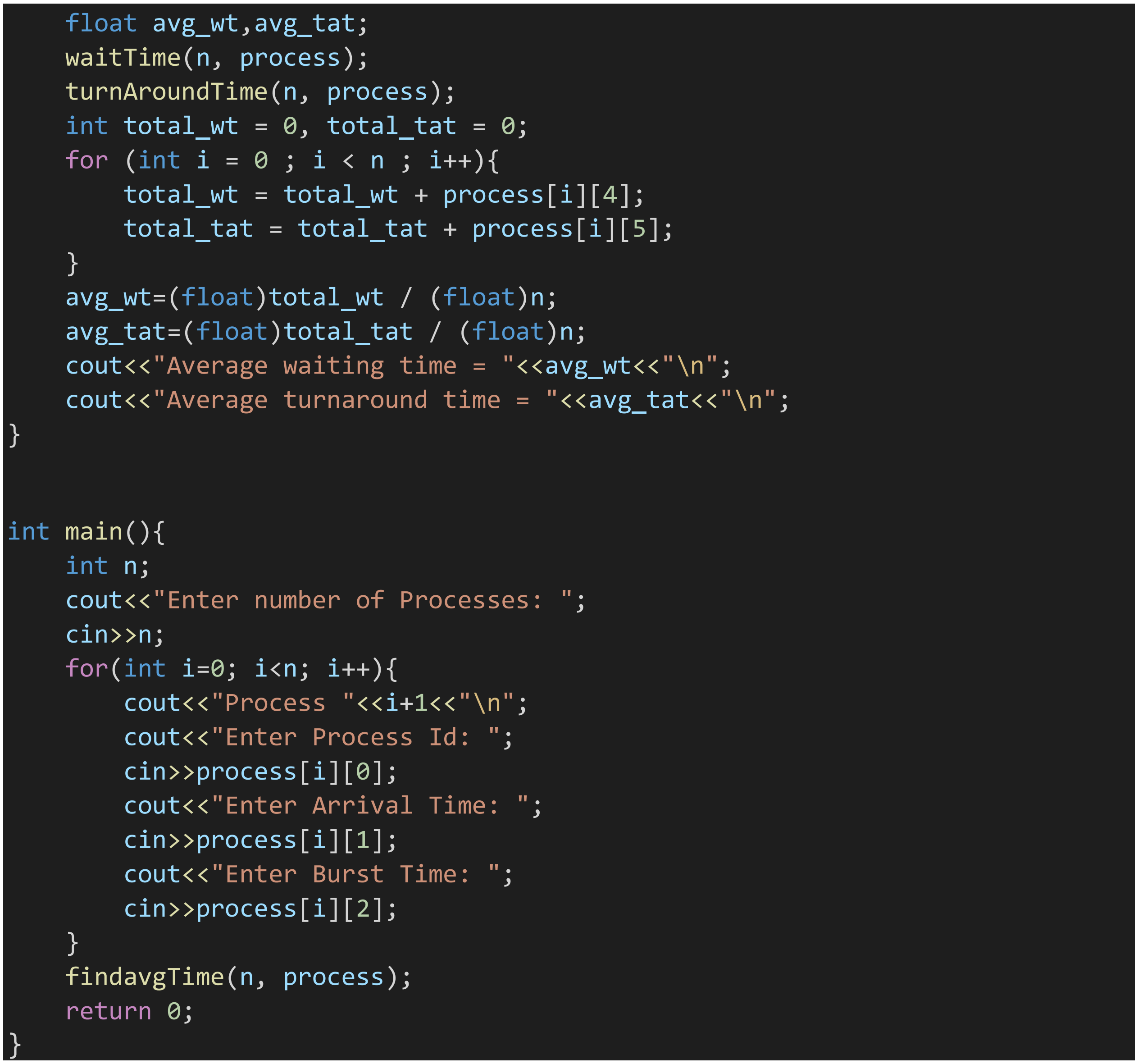
Example:

Process	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

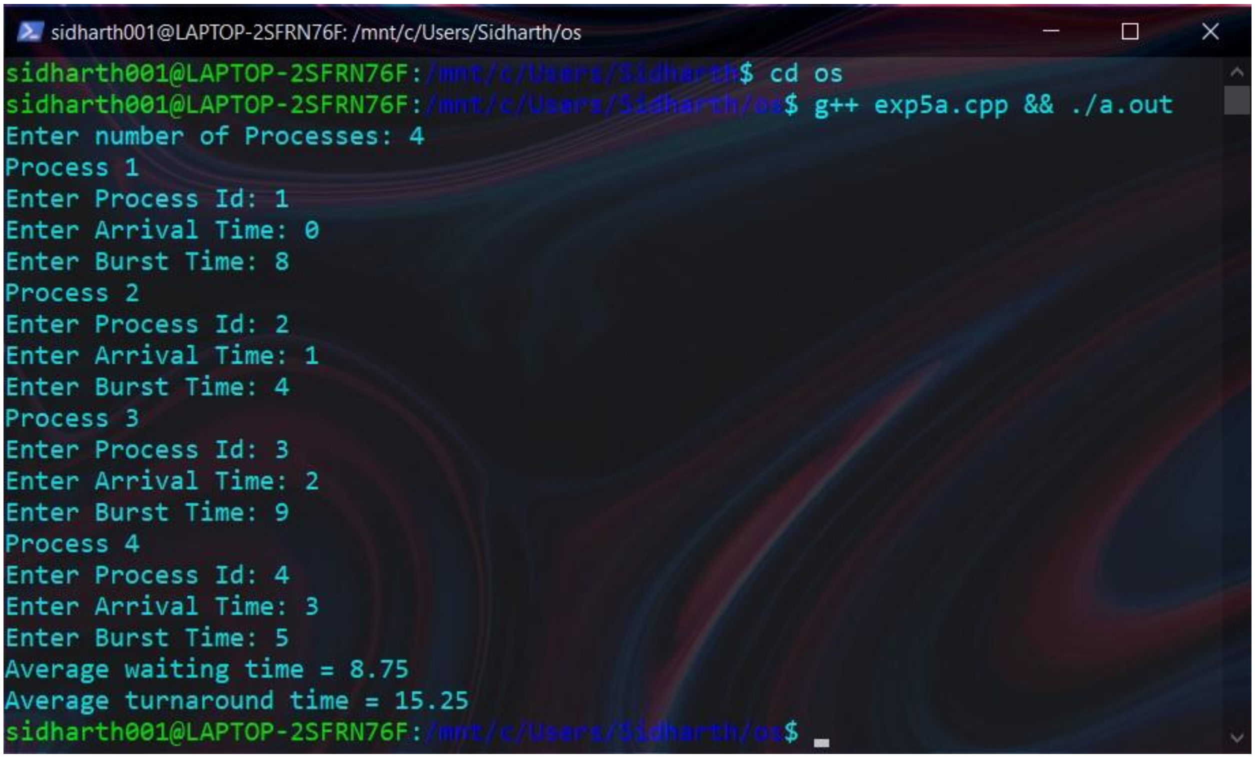
Code:

FCFS:

```
#include<iostream>
using namespace std;
int process[20][6];

void waitTime(int n, int process[][6]){
    for (int i = 1; i < n ; i++){
        process[i][3] = process[i-1][3] + process[i-1][2];
        process[i][4] = process[i][3] - process[i][1];
        if (process[i][4] < 0)  process[i][4] = 0;
    }
}

void turnAroundTime(int n, int process[][6]){
    for (int i=0; i<n; i++)  process[i][5] = process[i][2] + process[i][4];
}

void findavgTime(int n, int process[][6]){
```

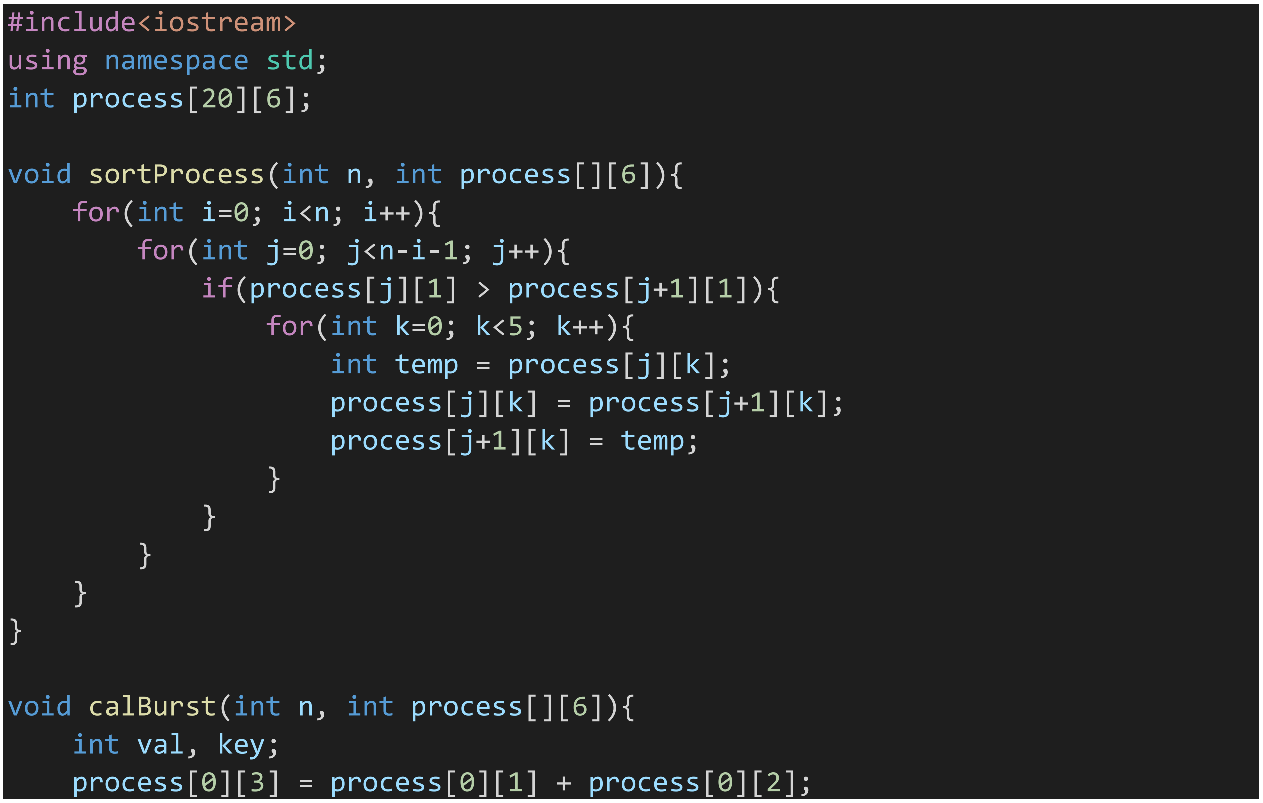


```cpp
    float avg_wt,avg_tat;
    waitTime(n, process);
    turnAroundTime(n, process);
    int total_wt = 0, total_tat = 0;
    for (int i = 0 ; i < n ; i++){
        total_wt = total_wt + process[i][4];
        total_tat = total_tat + process[i][5];
    }
    avg_wt=(float)total_wt / (float)n;
    avg_tat=(float)total_tat / (float)n;
    cout<<"Average waiting time = "<<avg_wt<<"\n";
    cout<<"Average turnaround time = "<<avg_tat<<"\n";
}


int main(){
    int n;
    cout<<"Enter number of Processes: ";
    cin>>n;
    for(int i=0; i<n; i++){
        cout<<"Process "<<i+1<<"\n";
        cout<<"Enter Process Id: ";
        cin>>process[i][0];
        cout<<"Enter Arrival Time: ";
        cin>>process[i][1];
        cout<<"Enter Burst Time: ";
        cin>>process[i][2];
    }
    findavgTime(n, process);
    return 0;
}
```

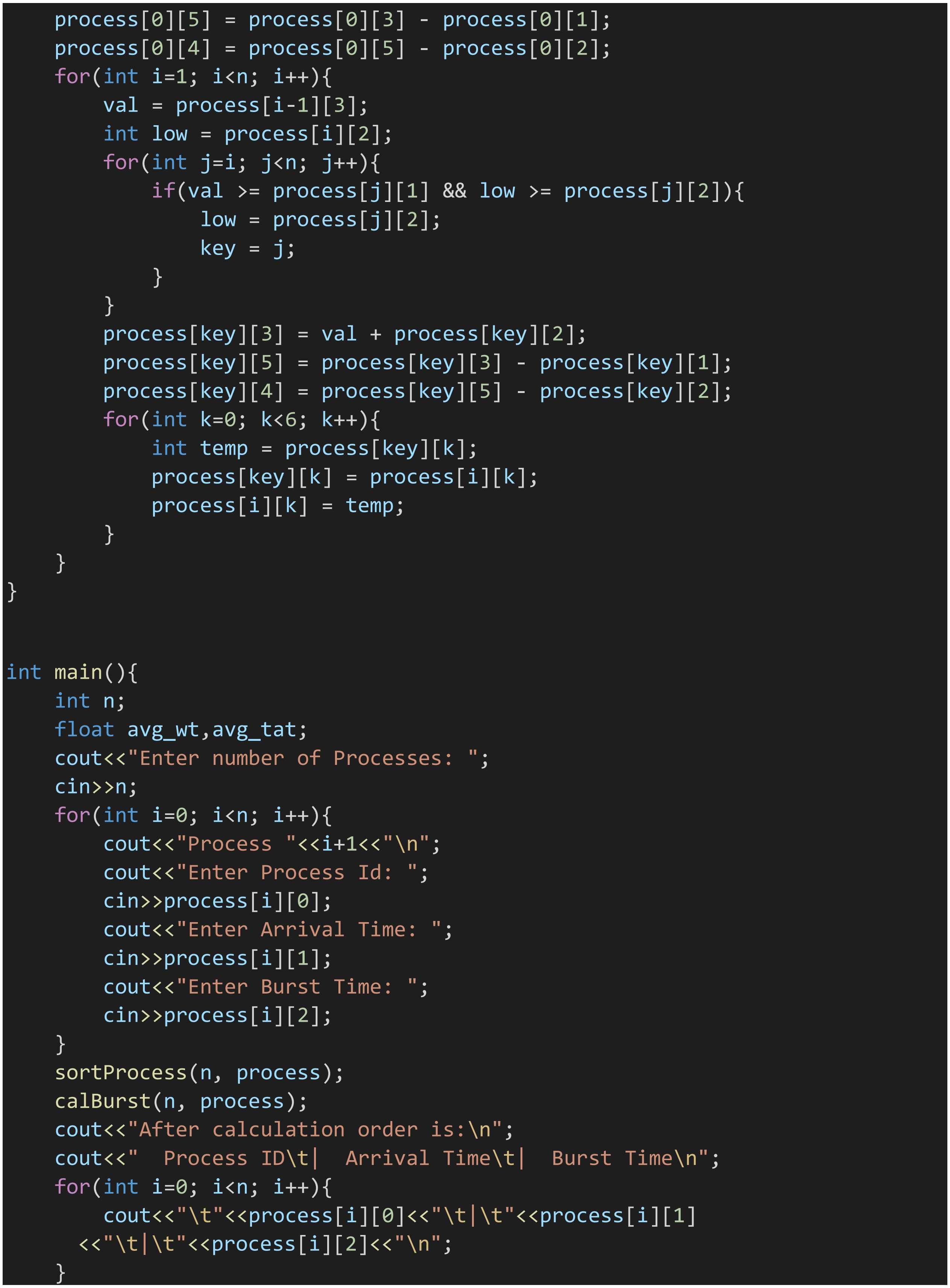


```
sidharth001@LAPTOP-2SFRN76F:/mnt/c/Users/Sidharth$ cd os
sidharth001@LAPTOP-2SFRN76F:/mnt/c/Users/Sidharth/os$ g++ exp5a.cpp && ./a.out
Enter number of Processes: 4
Process 1
Enter Process Id: 1
Enter Arrival Time: 0
Enter Burst Time: 8
Process 2
Enter Process Id: 2
Enter Arrival Time: 1
Enter Burst Time: 4
Process 3
Enter Process Id: 3
Enter Arrival Time: 2
Enter Burst Time: 9
Process 4
Enter Process Id: 4
Enter Arrival Time: 3
Enter Burst Time: 5
Average waiting time = 8.75
Average turnaround time = 15.25
sidharth001@LAPTOP-2SFRN76F:/mnt/c/Users/Sidharth/os$ _
```

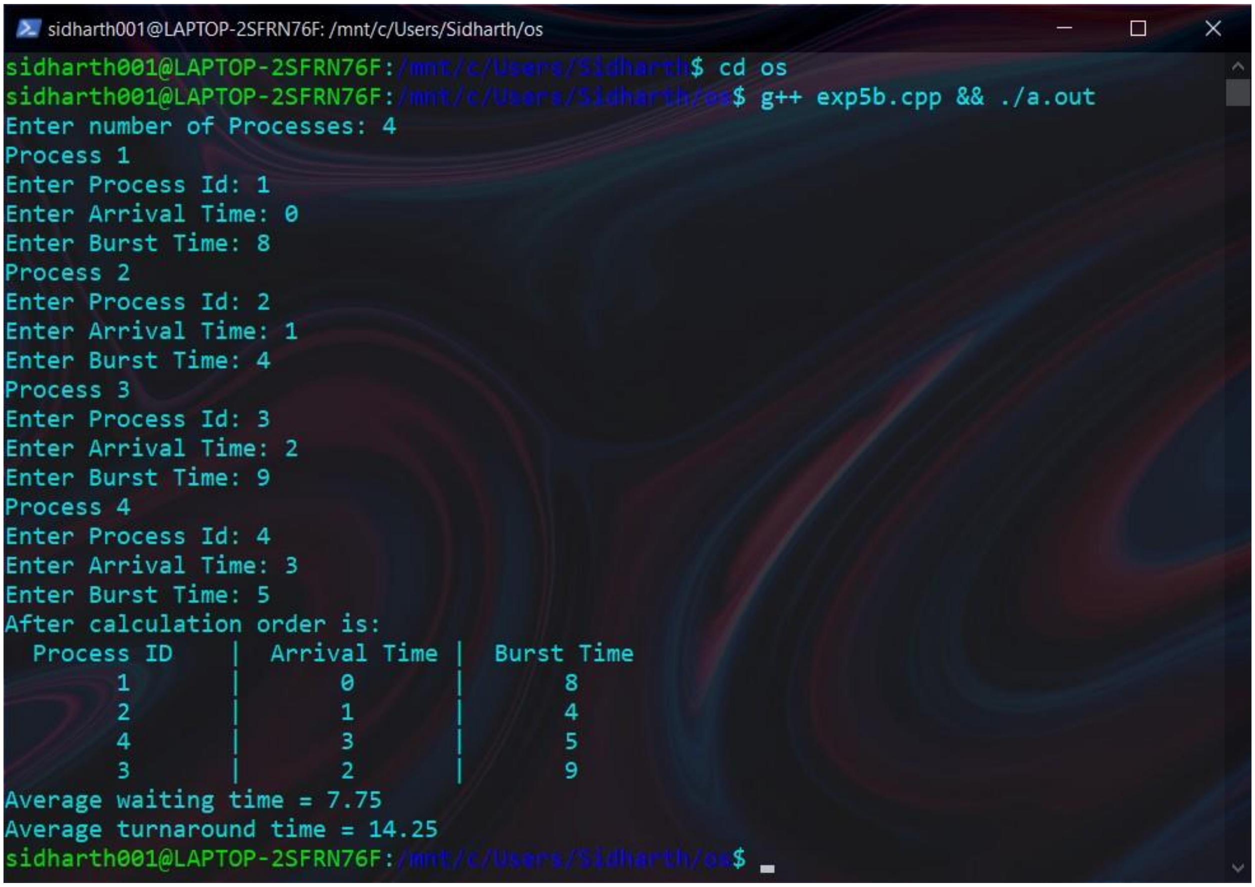
Non preemptive SJF:



```cpp
#include<iostream>
using namespace std;
int process[20][6];

void sortProcess(int n, int process[][6]){
    for(int i=0; i<n; i++){
        for(int j=0; j<n-i-1; j++){
            if(process[j][1] > process[j+1][1]){
                for(int k=0; k<5; k++){
                    int temp = process[j][k];
                    process[j][k] = process[j+1][k];
                    process[j+1][k] = temp;
                }
            }
        }
    }
}

void calBurst(int n, int process[][6]){
    int val, key;
    process[0][3] = process[0][1] + process[0][2];
```



```
    process[0][5] = process[0][3] - process[0][1];
    process[0][4] = process[0][5] - process[0][2];
    for(int i=1; i<n; i++){
        val = process[i-1][3];
        int low = process[i][2];
        for(int j=i; j<n; j++){
            if(val >= process[j][1] && low >= process[j][2]){
                low = process[j][2];
                key = j;
            }
        }
        process[key][3] = val + process[key][2];
        process[key][5] = process[key][3] - process[key][1];
        process[key][4] = process[key][5] - process[key][2];
        for(int k=0; k<6; k++){
            int temp = process[key][k];
            process[key][k] = process[i][k];
            process[i][k] = temp;
        }
    }
}


int main(){
    int n;
    float avg_wt,avg_tat;
    cout<<"Enter number of Processes: ";
    cin>>n;
    for(int i=0; i<n; i++){
        cout<<"Process "<<i+1<<"\n";
        cout<<"Enter Process Id: ";
        cin>>process[i][0];
        cout<<"Enter Arrival Time: ";
        cin>>process[i][1];
        cout<<"Enter Burst Time: ";
        cin>>process[i][2];
    }
    sortProcess(n, process);
    calBurst(n, process);
    cout<<"After calculation order is:\n";
    cout<<"  Process ID\t|  Arrival Time\t|  Burst Time\n";
    for(int i=0; i<n; i++){
        cout<<"\t"<<process[i][0]<<"\t|\t"<<process[i][1]
      <<"\t|\t"<<process[i][2]<<"\n";
    }
    float total=0;
    for(int i=0;i<n;i++)  total=total+process[i][4];
    avg_wt=(float)total/n;
    total=0;
    for(int i=0;i<n;i++)  total=total+process[i][5];
    avg_tat=(float)total/n;
    cout<<"Average waiting time = "<<avg_wt<<"\n";
    cout<<"Average turnaround time = "<<avg_tat<<"\n";
    return 0;
}
```



```
sidharth001@LAPTOP-2SFRN76F:/mnt/c/Users/Sidharth$ cd os
sidharth001@LAPTOP-2SFRN76F:/mnt/c/Users/Sidharth/os$ g++ exp5b.cpp && ./a.out
Enter number of Processes: 4
Process 1
Enter Process Id: 1
Enter Arrival Time: 0
Enter Burst Time: 8
Process 2
Enter Process Id: 2
Enter Arrival Time: 1
Enter Burst Time: 4
Process 3
Enter Process Id: 3
Enter Arrival Time: 2
Enter Burst Time: 9
Process 4
Enter Process Id: 4
Enter Arrival Time: 3
Enter Burst Time: 5
After calculation order is:
  Process ID    |  Arrival Time |  Burst Time
        1       |       0       |       8
        2       |       1       |       4
        4       |       3       |       5
        3       |       2       |       9
Average waiting time = 7.75
Average turnaround time = 14.25
sidharth001@LAPTOP-2SFRN76F:/mnt/c/Users/Sidharth/os$
```