# CS212: Assignment 1

Md Shabbir Jamal

Department of Computer Science and Engineering
BIT, Mesra, Ranchi
btech10026.20@bitmesra.ac.in

1. WAP to schedule process according to **F**CFS / First Come First Served scheduling algorithm

```
#include<iostream>
#include<algorithm>
#include<vector>
#include<string>
using namespace std;

//structure to store process detail
struct Process
{
    string name;
    int arrival_time;
    int burst_time;
};

//helpful in sorting the process by arrival time
bool a_t_sort(Process a,Process b)
{
    return a.arrival_time < b.arrival_time;
}

void input_process(vector<Process> &Proc)
{
    //taking input
    for(int i = 0;i<Proc.size();i++)
    {
        cout<<"p["<<i+1<<"] : ";
        Proc[i].name = to_string(i+1);

        cout<<"Arrival time : ";
        cin>>Proc[i].arrival_time;
        cout<<"       ";

        cout<<"Burst Time : ";
        cin>>Proc[i].burst_time;
        cout<<"\n";
    }
}

void Gantt_chart_n_Result(vector<Process> &Proc)
{
    //Gantt Chart
    sort(Proc.begin(),Proc.end(),a_t_sort);

    cout<<"\nGantt Chart : "<<"\n\n";
    vector<int> time;
     if(Proc[0].arrival_time != 0)
    {
        cout<<"|||";
        time.push_back(0);
    }
    time.push_back(Proc[0].arrival_time);
    for (int i=0; i<Proc.size(); i++)
    {
        time.push_back(Proc[i].burst_time) ;
    }
    for (int i=0; i<Proc.size(); i++)
    {
        cout<<" |||P["<< Proc[i].name << "]|||";
    }
    cout<<"|\n"<<time[0]<<"   ";
    for(int i = 1;i<time.size();i++)
    {
        time[i] = time[i] + time[i-1];
        cout<<time[i]<<"         ";
    }

    cout<<"\n\nResults : \n\n";

    //Waiting Time
    double waiting_time = 0.0;
    int i = 0;
    if(Proc[0].arrival_time != time[0])
    {
        i = 1;
    }
    int temp = 0;
    for( ; i < time.size() - 1 ;i++)
    {
        waiting_time = waiting_time - Proc[temp].arrival_time + time[i];
        temp++;
    }
    cout<<"Average Waiting Time : "<<waiting_time/Proc.size()<<"\n";

    //Turnaround Time
    double turnaround_time = 0.0;
    i = 1;
    if(Proc[0].arrival_time != time[0])
    {
        i = 2;
    }
    temp = 0;
    for( ; i < time.size();i++)
    {
        turnaround_time = turnaround_time - Proc[temp].arrival_time + time[i];
        temp++;
    }
    cout<<"Average Turaround Time : "<<turnaround_time/Proc.size()<<"\n";
}

int main()
{
    int n;
    cout<<"\t\tEnter Process Details : "<<"\n";
    cout<<"Enter the number of Process : ";
    cin>>n;

    //vector to store processes
    vector<Process> Proc(n);

    //input process detail
    input_process(Proc);

    //Calculate Gantt Chart and results
    Gantt_chart_n_Result(Proc);

    return 0;
}
```

### Output

```
Enter Process Details :
Enter the number of Process : 5
p[1] : Arrival time : 2
       Burst Time : 5

p[2] : Arrival time : 1
       Burst Time : 24
```

```
p[3] : Arrival time : 3
       Burst Time : 16

p[4] : Arrival time : 2
       Burst Time : 10

p[5] : Arrival time : 5
       Burst Time : 3


Gantt Chart :

||| |||P[2]||| |||P[1]||| |||P[4]||| |||P[3]||| |||P[5]||||
0   1         25         30         40         56         59

Results :

Average Waiting Time : 27.8
Average Turaround Time : 39.4
```