#include <iostream>

#include <algorithm>

using namespace std;

struct process

{

int arrival\_t ,burst\_t, priority ,proceessno;

}proc[50];

bool compare(process a, process b)

{

if(a.arrival\_t == b.arrival\_t)

{

a.priority < b.priority ;

}

else

{

return a.arrival\_t < b.arrival\_t ;

}

}

void GetWaitingTime (int wt[] , int n )

{

int service[50];

service[0]=0;

wt[0]=0;

for (int i = 1;

i < n;

i++)

{

service[i] = proc[i-1].burst\_t + service[i-1];

wt[i] = service[i] - proc[i].arrival\_t + 1;

if( wt[i] < 0)

{

wt[i]=0;

}

}

}

void GetTurnAroundTime(int tat[], int wt[], int n)

{

for(int i=0 ;

i<n ;

i++)

{

tat[i]=proc[i].burst\_t + wt[i];

}

}

void FINDGC(int n)

{

int wt[50],tat[50];

double wavg=0,tavg=0;

GetWaitingTime(wt, n);

GetTurnAroundTime(tat, wt, n);

int Stime[50], Ctime[50];

Stime[0] = 1 ;

Ctime[0] = Stime[0] + tat[0];

for(int i = 1 ;

i < n ;

i++)

{

Stime[i]= Ctime[i-1];

Ctime[i] = Stime[i] + tat[i] - wt[i];

}

cout<<"Process\_no\tStart\_time\tComplete\_time\tTurn\_Around\_Time\tWaiting\_Time"<<endl;

for(int i = 0 ;

i < n ;

i++)

{

wavg += wt[i];

tavg += tat[i];

cout << proc[i].proceessno << "\t\t" << Stime[i] << "\t\t"<<Ctime[i]<<"\t\t"<< tat[i]<<"\t\t\t"<<wt[i]<<endl;

}

cout<<"Average waiting time is : ";

cout<<wavg/(float)n<<endl;

cout<<"average turnaround time : ";

cout<<tavg/(float)n<<endl;

}

int main()

{

int n ;

cout << "Enter number of processes:- ";

cin >> n ;

int arrivaltime[n];

cout << "Enter Arrival Time For All Processes:- ";

for(auto &x : arrivaltime) cin >> x ;

int bursttime[n];

cout << "Enter Burst Time For All Processes:- ";

for(auto &x :bursttime) cin >> x ;

int priority[n];

cout << "Enter Priority For All Processes:- ";

for(auto &x : priority) cin >> x ;

for(int i=0;

i<n;

i++)

{

proc[i].arrival\_t=arrivaltime[i];

proc[i].burst\_t=bursttime[i];

proc[i].priority=priority[i];

proc[i].proceessno=i+1;

}

//Using inbuilt sort function

sort(proc,proc+n,compare);

FINDGC(n);

return 0;

}