#include <iostream>

#include <stdio.h>

#include <string>

using namespace std;

void FCFS(); //先来先服务

void SJF(); //短作业优先

void HRN(); //最高响应比

void input(); //输入函数

void operation(); //运算输出函数

int n; //全局变量作业输入量

int i=1;

double avturn; //平均周转时间

double avdqzz; //平均带权周转时间

double t=0.0,k=0.0; //全局变量用来表示总的周转时间和总的带权周转时间

struct HomeWork

{

int num; //作业号

double advtime; //到达时间

double runtime; //执行时间

double begintime; //开始时间

double endtime; //结束时间

double turntime; //周转时间

double dqzztime; //带权周转时间

double zgxy; //最高响应比

}HW[10];

void input()

{

cout<<"请输入要执行作业的数量："<<endl;

cin>>n;

for(int i=1;i<=n;i++)

{

cout<<"请输入第"<<i<<"个作业的数据："<<endl;

HW[i].num=i;

cout<<"到达时间：";

cin>>HW[i].advtime;

cout<<"执行时间：";

cin>>HW[i].runtime;

}

}

void operation()

{

for(int i=1;i<=n;i++)

{

if(i==1)

HW[i].begintime=HW[i].advtime; //如果是第一个作业开始时间就是到达时间

else

HW[i].begintime=HW[i-1].endtime;//如果不是，则作业的开始时间是前一个的结束时间

HW[i].endtime=HW[i].begintime+HW[i].runtime;

HW[i].turntime=HW[i].endtime-HW[i].advtime;

HW[i].dqzztime=HW[i].turntime/HW[i].runtime;

}

for(i=1;i<=n;i++)

{

t=t+HW[i].turntime;

k=k+HW[i].dqzztime;

}

avturn=t/n;

avdqzz=k/n;

cout<<"作业序号到达时间执行时间开始时间结束时间周转时间带权周转时间"<<endl;

for(i=1;i<=n;i++)

cout<<" "<<HW[i].num<<" \t"<<HW[i].advtime<<" \t"<<HW[i].runtime<<" \t"<<HW[i].begintime<<" \t"<<HW[i].endtime<<" \t"<<HW[i].turntime<<" \t"<<HW[i].dqzztime<<endl;

cout<<"平均周转时间："<<avturn<<endl;

cout<<"平均带权周转时间："<<avdqzz<<endl;

}

void FCFS()

{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*先来先服务（FSFC）算法矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

operation();

t=0.0;

k=0.0;

}

void SJF()

{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*短作业优先（SJF）算法矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

int no; //作业号

double tjtime; //到达时间

double yxtime; //执行时间

for(int i=2;i<=n;i++)

for(int j=i+1;j<=n;j++)

{

if(HW[i].runtime>HW[j].runtime)

{

no=HW[i].num;

HW[i].num=HW[j].num;

HW[j].num=no; //作业号互换

tjtime=HW[i].advtime;

HW[i].advtime=HW[j].advtime;

HW[j].advtime=tjtime; //作业到达时间互换

yxtime=HW[i].runtime;

HW[i].runtime=HW[j].runtime;

HW[j].runtime=yxtime; //作业执行时间互换

}

}

operation();

t=0.0;

k=0.0;

}

void HRN()

{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*最高响应比优先（HRN）算法矩阵\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

double tjtime,yxtime;

int no;

for(int i=2;i<=n;i++)

{

if(HW[i].advtime<HW[i-1].endtime)//判断第二个作业是否是在第一个作业完成前到达的

HW[i].zgxy=(HW[i-1].endtime-HW[i].advtime)/HW[i].runtime +1;

for(int j=i+1;j<=n;j++)

{

if(HW[j].advtime<HW[i].endtime)//判断后面作业是否是前一个作业完成前到达的

HW[j].zgxy=(HW[j].begintime-HW[j].advtime)/HW[j].runtime +1;

if(HW[i].zgxy<HW[j].zgxy)

{

no=HW[i].num;

HW[i].num=HW[j].num;

HW[j].num=no; //作业号互换

tjtime=HW[i].advtime;

HW[i].advtime=HW[j].advtime;

HW[j].advtime=tjtime; //作业到达时间互换

yxtime=HW[i].runtime;

HW[i].runtime=HW[j].runtime;

HW[j].runtime=yxtime; //作业执行时间互换

}

}

}

operation();

t=0.0;

k=0.0;

}

void main()

{

input();

FCFS();

SJF();

HRN();

}