LRTF

#include <iostream>

#include <queue>

#include <iomanip>

using namespace std;

struct process{

int id;

int atm;

int bt;

int ct;

int tat;

int wt;

int st;

};

struct compare{

bool operator()(process const& p1, process const& p2)

{

if(p1.atm != p2.atm){

return p1.atm > p2.atm;}

else{

return p1.id>p2.id;

}

}

};

struct compareID{

bool operator()(process const& p1, process const& p2)

{

return p1.id>p2.id;

}

};

int main()

{

int n;

cout<<"Enter the number of process : ";

cin>>n;

process pr[n];

int p[n];

int pa[n];

int pb[n];

cout<<"Enter the arrival time of all processes : ";

for(int i=0;i<n;i++)

{

cin>>pa[i];

p[i]=i+1;

pr[i].atm=pa[i];

pr[i].id=p[i];

}

cout<<"Enter the burst time of all processes : ";

for(int i=0;i<n;i++)

{

cin>>pb[i];

pr[i].bt=pb[i];

}

priority\_queue<process,vector<process>,compare>Q;

for(int i=0;i<n;i++)

{

Q.push(pr[i]);

}

int i=0;

while(!Q.empty())

{

pr[i]=Q.top();

Q.pop();

i++;

}

for(int i=0;i<n;i++)

{

pb[i]=pr[i].bt;

}

int c=0;

for(int i=0;i<n;i++)

{

c+=pr[i].bt;

}

int done[n]={0};

int cp=pr[0].atm;

for(int i=0;i<n;i++)

{

pr[i].ct=pr[0].atm;

}

while(c--)

{

int m=0;

int flag;

for(int i=0;i<n;i++)

{

if(pr[i].atm<=cp&&pr[i].bt>m&&done[i]==0&&pr[i].bt>0)

{

m=pr[i].bt;

flag=i;

}

}

cp+=1;

for(int i=0;i<n;i++)

{

if(done[i]==0)

{

pr[i].ct+=1;

}

}

pr[flag].bt-=1;

if(pr[flag].bt==0)

{

done[flag]=1;

}

if(done[flag]==1)

{

pr[flag].tat=pr[flag].ct-pr[flag].atm;

pr[flag].wt=pr[flag].tat-pb[flag];

}

}

for(int i=0;i<n;i++)

{

pr[i].bt=pb[i];

}

priority\_queue<process,vector<process>,compareID>Q1;

for(int i=0;i<n;i++)

{

Q1.push(pr[i]);

}

int j=0;

while(!Q1.empty())

{

pr[j]=Q1.top();

Q1.pop();

j++;

}

cout<<"P\_No\t"<<"AT\t"<<"BT\t"<<"CT\t"<<"TAT\t"<<"WT\t"<<endl;

for(int i=0;i<n;i++)

{

cout<<pr[i].id<<"\t"<<pr[i].atm<<"\t"<<pr[i].bt<<"\t"<<pr[i].ct<<"\t"<<pr[i].tat<<"\t"<<pr[i].wt<<endl;

}

cout<<endl;

float avg\_tat=0;

float avg\_wt=0;

for(int i=0;i<n;i++)

{

avg\_tat+=pr[i].tat;

avg\_wt+=pr[i].wt;

}

avg\_tat/=n;

avg\_wt/=n;

cout<<fixed<<setprecision(2)<<"Average turn Around Time is : "<<avg\_tat<<endl;

cout<<"Average Waiting Time is : "<<avg\_wt<<endl;

return 0;

}