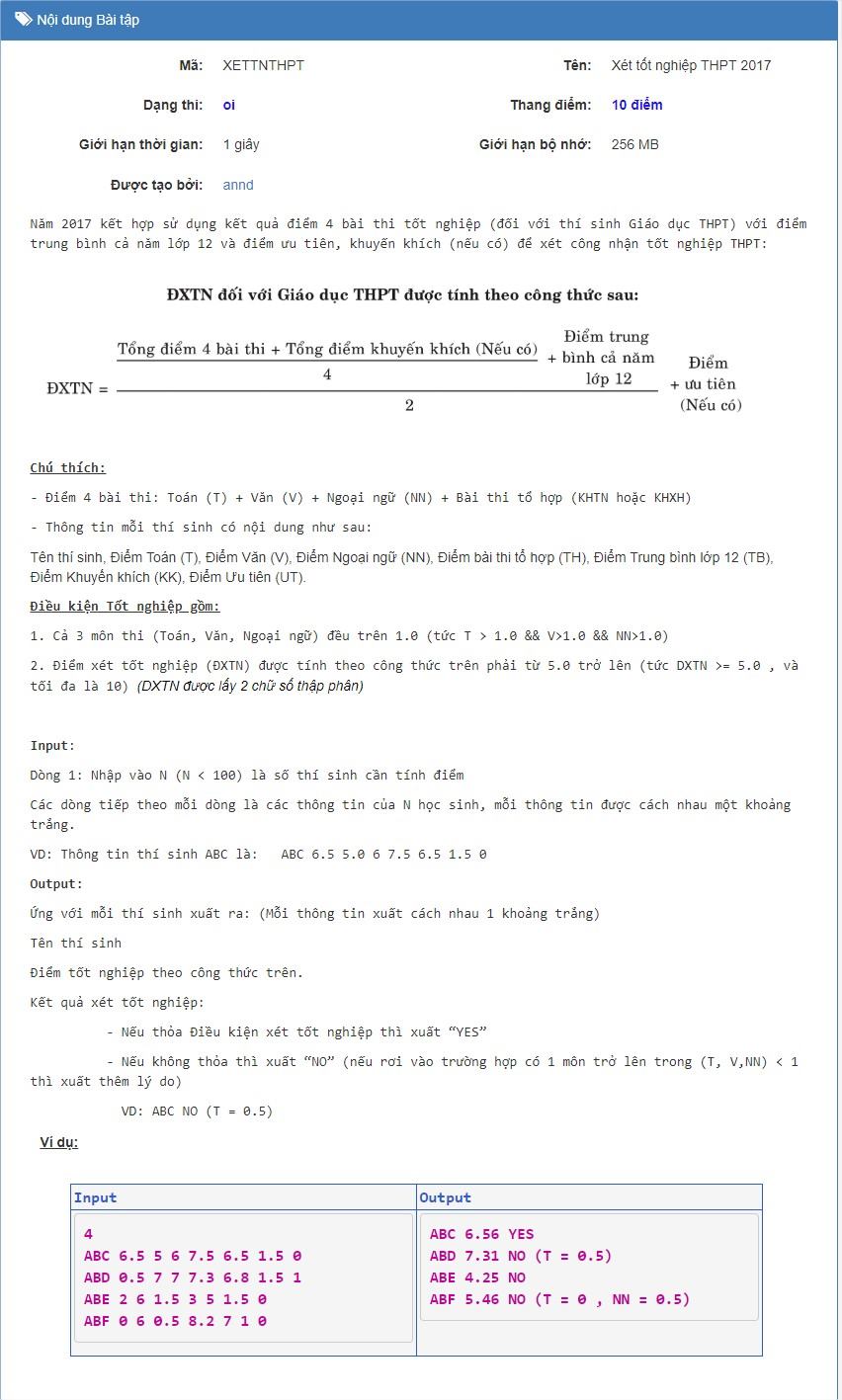
# Điểm thi tốt nghiệp



#include<bits/stdc++.h>

#define MAX 10000

//khai bao bien

struct thisinh{

string ten;

float toan,van,nn;//toan,van,ngoaingu

float th,tb,kk,ut;//tohop,trungbinh12,khuyenkhich,uutien

float dtn;

};

struct thisinh TS[MAX];

int n;//so luong thi sinh

float xtn[MAX][4];

void readdata(){

cin>>n;

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

cin>>TS[i].ten>>TS[i].toan>>TS[i].van>>TS[i].nn>>TS[i].th>>TS[i].tb>>TS[i].kk>>TS[i].ut;

}

//luu cac gia tri vao mang 2 chieu: dtn, toan, van, ngoai ngu cho tung thi sinh

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

TS[i].dtn=(float)((TS[i].toan + TS[i].van + TS[i].nn + TS[i].th)/4+TS[i].tb)/2+TS[i].ut;

xtn[i][0]=TS[i].dtn;

xtn[i][1]=TS[i].toan;

xtn[i][2]=TS[i].van;

xtn[i][3]=TS[i].nn;

}

//in ra ma tran xtn

/\*

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

for(int j=0; j<4; j++){

cout<<xtn[i][j]<<" ";

}

cout<<endl;

}

\*/

}

void ketqua(){

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

if(TS[i].dtn <5){

cout<<TS[i].ten<<" "<<TS[i].dtn<<" NO"<<endl;

}

else{

if(TS[i].dtn >= 5){

int dem=0;

int diemliet[3]={0,0,0};//quy dinh ko co mon nao bi liet = 0 (toan, van, ngoai ngu)

for(int j=1; j<4; j++){

if(xtn[i][j] < 1){

dem++;

//luu vao mang danh dau mon co diem liet

diemliet[j-1]=1;

}

}

//het for khong co diem nao <=1

if(dem==0){

cout<<TS[i].ten<<" "<<TS[i].dtn<<" YES"<<endl;

}

else{//dem!=0 nghia la co diem liet

cout<<TS[i].ten<<" "<<TS[i].dtn<<" NO (";

for(int k=0; k<3; k++){

if(diemliet[k]==1){

switch (k)

{

case 0:

cout<<"T = "<<TS[i].toan;

break;

case 1:

cout<<" V = "<<TS[i].van;

break;

case 2:

cout<<" NN = "<<TS[i].nn;

break;

default:

break;

}

}

}

cout<<")"<<endl;

}

}

}

}

}

//chuong trinh chinh

int main(){

ios::sync\_with\_stdio(0);

cin.tie(0);

cout.tie(0);

readdata();

ketqua();

return 0;

}

# Quy hoạch động

Khi nào thì dùng thuật toán quy hoạch động

Khi nào thì chúng ta cần đến quy hoạch động? Đó là một câu hỏi rất khó trả lời. Không có một công thức nào cho các bài toán như vậy.

Tuy nhiên, có một số tính chất của bài toán mà bạn có thể nghĩ đến quy hoạch động. Dưới đây là hai tính chất nổi bật nhất trong số chúng:

*Bài toán có các bài toán con gối nhau.*

*Bài toán có cấu trúc con tối ưu.*

Thường thì một bài toán có đủ cả hai tính chất này, chúng ta có thể dùng quy hoạch động được. Một câu hỏi rất thú vị là không dùng quy hoạch động có được không? Câu trả lời là có, nhưng nếu bạn đi thi code, bạn trượt là cái chắc. Để hiểu rõ hơn, chúng ta sẽ tìm hiểu từng tính chất một trong những phần dưới đây