#include<bits/stdc++.h>

using namespace std;

class node{

public:

string name;

int files;

vector<node\*> child;

};

node\* insert\_node(vector<pair<string,int> > &v){

stack<node\*> st;

node\* root;

for(int i=0;i<v.size();i++){

if(v[i].second==-1){

st.pop();

}

else{

node\* temp = new node();

temp->name = v[i].first;

temp->files = v[i].second;

if(st.size()>0){

st.top()->child.push\_back(temp);

}

else{

root = temp;

}

st.push(temp);

}

}

return root;

}

void display(node\* root){

if(root==NULL){

return;

}

cout<<root->name<<"->"<<root->files<<endl;

for(auto it: root->child){

display(it);

}

}

int max\_files(node\* root, int max\_){

if(root==NULL){

return -1;

}

for(auto it: root->child){

max\_ = max(max\_,max\_files(it,max\_));

}

max\_ = max(max\_,root->files);

return max\_;

}

int main(){

int n;

cin>>n;

vector<pair<string,int> > v(n); // {folder\_name, files\_number}

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

cin>>v[i].first;

cin>>v[i].second;

}

node\* root = insert\_node(v);

display(root);

cout<<max\_files(root,-100)<<endl;

}

// sample input

// 22

// root 10

// C 20

// F1 50

// N -1

// F6 60

// N -1

// N -1

// D 30

// F2 70

// N -1

// F3 80

// FF31 110

// N -1

// FF32 120

// N -1

// N -1

// F4 90

// N -1

// N -1

// E 40

// F5 100

// N -1

/\* root,10

C,20 D,30 E,40

F1,50 F6,60 F2,70 F3,80 F4,90 F5,100

FF31,110 FF32,120

\*/