

## 树状结构查询

### 题目描述：

通常使用多行的节点、父节点表示一棵树，比如

西安 陕西

陕西 中国

江西 中国

中国 亚洲

泰国 亚洲

输入一个节点之后，请打印出来树中他的所有下层节点

### 输入描述：

第一行输入行数，下面是多行数据，每行以空格区分节点和父节点

接着是查询节点

### 输出描述：

输出查询节点的所有下层节点。以字典序排序

### 补充说明：

树中的节点是唯一的，不会出现两个节点，是同一个名字

### 示例 1

#### 输入：

```
5
b a
c a
d c
e c
f d
c
```

#### 输出：

```
d
e
f
```

```
#include <iostream>
```

```
#include<unordered_map>
```

```

#include<vector>

#include <algorithm>

#include<set>

using namespace std;

void findDescendants(const unordered_map<string,vector<string>>&tree,const
string&node,set<string>&descendants){

    if(tree.find(node)==tree.end())

        return;

    for(const string&child:tree.at(node)){

        descendants.insert(child);

        findDescendants(tree,child,descendants);

    }

}

int main() {

    int n;

    cin>>n;

    unordered_map<string, vector<string>>tree;

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

        string node,parent;

        cin>>node>>parent;

```

```
    tree[parent].push_back(node);  
}  
  
string query;  
  
cin>>query;  
  
set<string>descendants;  
  
findDescendants(tree, query, descendants);  
  
for(const string&child:descendants){  
    cout<<child<<endl;  
}  
  
return 0;  
}
```