

题目描述：

某文件系统中存在 N 个目录，每个目录都有一个独一无二的 ID。每个目录只有一个父目录，但每个父目录下可以有零个或者多个子目录，目录结构呈树状结构。

假设，根目录的 ID 为 0，且根目录没有父目录，其他所有目录的 ID 用唯一的正整数表示，并统一编号。

现给定目录 ID 和其父目录 ID 的对应父子关系表[子目录 ID，父目录 ID]，以及一个待删除的目录 ID，请计算并返回一个 ID 序列，表示因为删除指定目录后剩下的所有目录，返回的 ID 序列以递增序输出。

注意：

- 1、被删除的目录或文件编号一定在输入的 ID 序列中；
- 2、当一个目录删除时，它所有的子目录都会被删除。

输入描述：

输入的第一行为父子关系表的长度 m ；接下来的 m 行为 m 个父子关系对；最后一行为待删除的 ID。序列中的元素以空格分割，参见样例。

输出描述：

输出一个序列，表示因为删除指定目录后，剩余的目录 ID。

补充说明：

示例 1

输入：

```
5
8 6
10 8
6 0
20 8
2 6
8
```

输出：

```
2 6
```

说明：

目录结构如下所示：

```
    6
   /  \
  2    8
     /  \
    10  20
```

删除目录 8，同时它的子目录 10 也被删除，剩余 2 和 6 两个目录。

```
m = int(input())
rlts = [list(map(int, input().split())) for _ in range(m)]
rm = int(input())
```

```
def dfs(tree, node, rm, res):
    if tree.get(node) is not None:
        children = tree[node]
        for child in children:
            if child !=rm:
                res.append(child)
                dfs(tree, child, rm, res)
def answer():
    tree = {}
    for child, father in rlts:
        if tree.get(father) is None:
            tree[father] = []
        tree[father].append(child)
    if rm == 0:
        return ""
    res = []
    dfs(tree, 0, rm, res)
    res.sort()
    return " ".join(map(str, res))
print(answer())
```