

python-递归-一个文件目录的数据格式为：目录 id

题目描述：

一个文件目录的数据格式为：目录 id，本目录中文件大小，(子目录 id 列表)。其中目录 id 全局唯一，取值范围[1,200]，本目录中文件大小范围[1,1000]，子目录 id 列表个数[0,10]

例如：1 20 (2,3) 表示目录 1 中文件总大小是 20，有两个子目录，id 分别是 2 和 3

现在输入一个文件系统中所有目录信息，以及待查询的目录 id，返回这个目录和及该目录所有子目录的大小之和。

输入描述：

第一行为两个数字 M,N，分别表示目录的个数和待查询的目录 id， $1 \leq M \leq 100$ ， $1 \leq N \leq 200$

接下来 M 行，每行为 1 个目录的数据：目录 id 本目录中文件大小 (子目录 id 列表)，子目录列表中的子目录 id 以逗号分隔。

输出描述：

待查询目录及其子目录的大小之和

补充说明：

示例 1

输入：

3 1

3 15 ()

1 20 (2)

2 10 (3)

输出：

45

说明：

目录 1 大小为 20，包含一个子目录 2（大小为 10），子目录 2 包含一个子目录 3（大小为 15），总的大小为 $20+10+15=45$ 。

示例 2

输入：

4 2

4 20 ()

5 30 ()

2 10 (4,5)

1 40 ()

输出：

60

说明：

目录 2 包含 2 个子目录 4 和 5，总的大小为 $10+20+30 = 60$

`x,y = map(int, input().split())`

`child = {}`

`cap = {}`

```

for _ in range(x):
    father_id, father_cap, child_str = input().split()
    child[father_id] = []
    cap[father_id] = int(father_cap)

    if len(child_str) > 2:
        child[father_id].extend(child_str[1:-1].split(","))

def getResout(tar):
    ansd = 0

    stack = [tar]

    while len(stack) > 0:
        id = stack.pop()
        if cap.get(id) is None:
            continue
        ansd += cap[id]
        stack.extend(child[id])

    return ansd

print(getResout(str(y)))

```