

## Contest 2

题目名称	two	bracket	sum
源程序文件名	two.pas/c/cpp	bracket.pas/c/cpp	sum.pas/c/cpp
输入文件名	two.in	bracket.in	sum.in
输出文件名	two.out	bracket.out	sum.out
时间限制	4s	1s	1s
内存限制	512MB	512MB	512MB
题目类型	传统型	传统型	传统型
是否打开-O2	是	是	是

# 1 two

## 1.1 Description

你有两棵有根树，每棵各有  $n$  个顶点。让我们用整数 1 到  $n$  给每棵树的顶点编号。两棵树的根都是顶点 1。第一棵树的边都是蓝色，第二棵树的边都是红色。简明起见，我们称第一棵树是蓝色的，以及第二棵树是红色的。

当满足下面的两个条件下，我们认为边  $(x, y)$  有害于边  $(p, q)$ ：

- 1.边  $(x, y)$  的颜色不同于边  $(p, q)$ 。
- 2.考虑与边  $(p, q)$  颜色相同的树，编号为  $x, y$  的两个顶点中有且只有一个同时在顶点  $p$  的子树与顶点  $q$  的子树里。

现在告诉你，在阶段 1，有恰好一条蓝色的边被删除了，

而在阶段  $i$ ，若我们删除了边  $(u_1, v_1), (u_2, v_2), \dots, (u_k, v_k)$ 。

那么在阶段  $i+1$  我们要删除的所有满足以下条件的边  $(x, y)$ ：

- 1.边  $(x, y)$  未被删除。
- 2.存在一个  $i (i \leq k)$  使得边  $(x, y)$  有害于  $(u_i, v_i)$ 。

当某个阶段没有删除任何边时，则整个过程结束，你需要回答，每个阶段哪些边将被删除。

注意，有害边的定义只依赖于开始删边之前的初始就拥有的两棵有根树。

## 1.2 Task

### 1.2.1 Input

输入第一行为整数  $n$ ，表示两棵树的顶点数目。

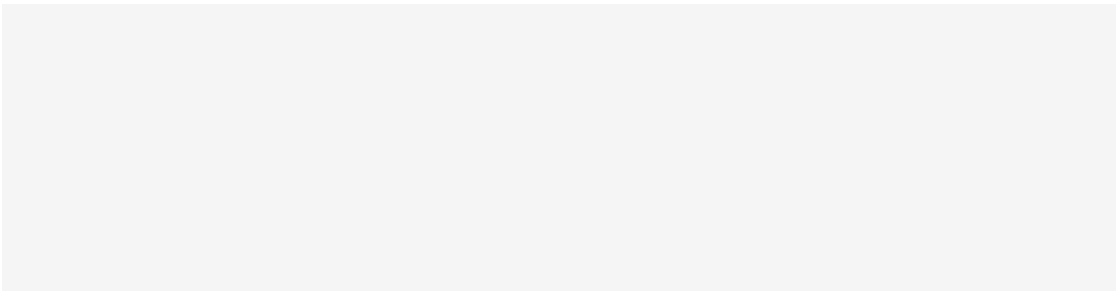
接下来的一行包含  $n-1$  个正整数  $a_2, a_3, \dots, a_n (1 \leq a_i \leq n; a_i \neq i)$ ，描述第一棵树的边。数字  $a_i$  意味着第一棵树有一条边连接顶点  $a_i$  和顶点  $i$ 。

接下来的一行包含  $n-1$  个正整数  $b_2, b_3, \dots, b_n (1 \leq b_i \leq n; b_i \neq i)$ ，描述第二棵树的边。数字  $b_i$  意味着第二棵树有一条边连接顶点  $b_i$  和顶点  $i$ 。

接下来的再一行包含一个整数  $idx (1 \leq idx < n)$  表示在第一阶段中删除的蓝树的边的编号。我们让每棵树的每条边按照他们在输入中的前后顺序从 1 到  $n-1$  编号。

### 1.2.2 Output

对于每个阶段输出两行。如果这个阶段删除的边是蓝色的，那么对应这一阶段的两行中，第一行必须为单词 **Blue**，否则为单词 **Red**。对应的第二行包含所有此阶段删除的边的编号，按数字递增顺序输出。



## 1.3 Sample

### 1.3.1 Input

5

1 1 1 1

4 2 1 1

3

### 1.3.2 Output

Blue

3

Red

1 3

Blue

1 2

Red

2

## 1.4 Constraint

对于 30% 的数据， $n \leq 100$ 。

对于 60% 的数据， $n \leq 1000$ 。

对于 100% 的数据， $n \leq 200000$ 。

## 2 bracket

### 2.1 Description

给定一棵有  $n$  个节点的无根树，每个节点上是一个字符，要么是(，要么是)。  
定义  $S(x, y)$  为从  $x$  开始沿着最短路走到  $y$ ，将沿途经过的点上的字符依次连起来得到的字符串。

合法括号序定义如下：

- 1, ()是合法的。
- 2, 若  $A$ ，合法，则 $(A)$ 也合法。
- 3, 若  $A$ ,  $B$  分别合法，则  $AB$  也合法。

函数  $f(x, y)$  等于对  $S(x, y)$  进行划分，使得每一个部分都是合法括号序，能得到的最大的段数，比如 $(())()$ 的最大段数为 3， $((()))()$ 的最大段数为 2。

特别的，如果  $S(x,y)$ 本身并不是合法括号序，则  $f(x,y)=0$ 。

$m$  次询问，每次输入一个  $k$ ，查询有多少点对的  $f$  值为  $k$ 。

### 2.2 Task

#### 2.2.1 Input

第一行一个整数  $n$ ，表示节点数。

接下来  $n-1$  行每行两个整数  $x,y$ ,描述一条边。

接下来  $n$  行，每行一个字符(或)，其中第  $i$  行表示  $i$  号节点的字符。

接下来一行一个整数  $m$ ，表示询问个数。

接下来  $m$  行，每行一个整数  $k$ ，表示一个询问。

#### 2.2.2 Output

输出共  $m$  行，每行一个整数表示有多少对  $x, y$  满足  $f(x,y)=k$ 。

### 2.3 Sample

#### 2.3.1 Input

```
6
1 2
2 6
4 2
3 4
1 5
)
(
)
)
(
)
```

3  
1  
2  
3

### 2.3.2 Output

4  
2  
0

## 2.4 Constraint

对于 10%的数据， $n, m \leq 100$

对于 30%的数据， $n, m \leq 5000$

对于另 30%的数据， $m \leq 10$

对于 100%的数据， $n, m \leq 50000$

## 3 sum

### 3.1 Description

请你取出集合 $\{1, 2, \dots, n\}$ 的一个子集，使得其中的元素两两互质，并最大化子集中的元素之和，你只需要输出这个子集中的元素之和即可。

如  $\text{ans}(10)=30$ ，此时的子集为 $\{1, 5, 7, 8, 9\}$ 。

### 3.2 Task

#### 3.2.1 Input

一行，一个正整数  $n$

#### 3.2.2 Output

一行一个正整数，表示答案

### 3.3 Sample

#### 3.3.1 Input

10

#### 3.3.2 Output

30

### 3.4 Constraint

对于 20% 的数据， $1 \leq n \leq 30$

对于 50% 的数据， $1 \leq n \leq 200$

对于 80% 的数据， $1 \leq n \leq 800$

对于 100% 的数据， $1 \leq n \leq 200000$