



朋友 (Friend)

我们建立了一个由编号为 $0, \dots, n-1$ 的 n 个人组成的社交网络。网络中的有些人会成为朋友。如果 X 号人成为 Y 号人的朋友，则 Y 号人同时也会成为 X 号人的朋友。

这些人将通过 n 个阶段加入这个网络，阶段也编号为 0 至 $n-1$ 。第 i 号人在第 i 个阶段加入。在阶段 0 ， 0 号人加入网络并成为唯一的人。此后 $n-1$ 个阶段的各个阶段，都有一个人会被主持人加入到网络中，而这个主持人可以是已在网络中的任何一个人。在阶段 i 中 ($0 < i < n$)，该阶段的主持人可以用如下三种方式之一把第 i 号人加入到网络中：

- *IAmYourFriend* 将第 i 号人仅变成主持人的朋友。
- *MyFriendsAreYourFriends* 将第 i 号人变成主持人当前的每一个朋友的朋友。注意，这种方式不会将第 i 号人变成主持人的朋友。
- *WeAreYourFriends* 将第 i 号人变成主持人的朋友，同时也变成主持人当前的每一个朋友的朋友。

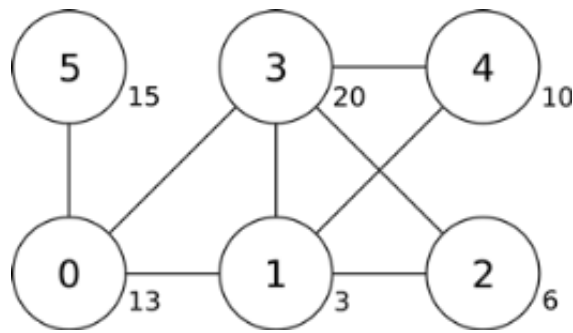
在建立此网络之后，我们想挑选一个调查的样本，也就是说要从网络中选择一组人。由于朋友之间通常拥有相似的兴趣，因此样本不应包含任何一对互为朋友的人。每个人都会有一个调查的可信度，表示为一个正整数，而我们想要找出一个可信度总和最大的样本。

例子

阶段	主持人	方式	被加入的朋友关系
1	0	IAmYourFriend	(1, 0)
2	0	MyFriendsAreYourFriends	(2, 1)
3	1	WeAreYourFriends	(3, 1), (3, 0), (3, 2)
4	2	MyFriendsAreYourFriends	(4, 1), (4, 3)
5	0	IAmYourFriend	(5, 0)

最初在网络中只包含 0 号人。阶段 1 的主持人 (0 号人) 用 *IAmYourFriend*方式去邀请 1 号人，因此他们成为朋友。阶段 2 的主持人 (同样是 0 号人) 用

*MyFriendsAreYourFriends*方式去邀请 2 号人，这样使 1 号人 (主持人的唯一朋友) 成为 2 号人的唯一朋友。阶段 3 的主持人 (1 号人) 用 *WeAreYourFriends*方式加入 3 号人，这样使 3 号人变成 1 号人 (主持人)、 0 号人和 2 号人 (主持人的朋友) 的朋友。阶段 4 和 5 都如上表所示。最后形成的网络如下图所示，其中圆圈内的数字表示人的编号，而圆圈旁的数值表示这个人的可信度。由 3 号人和 5 号人组成的样本的调查可信度总和为 $20 + 15 = 35$ ，这是所有可能的可信度总和的最大值。



任务

给定各阶段的描述以及每个人的可信度值，请找出一个可信度总和最大的样本。你只需要实现函数 `findSample`。

- `findSample(n, confidence, host, protocol)`
 - `n`: 人数.
 - `confidence`: 大小为 n 的数组；`confidence[i]` 表示第 i 号人的可信度。
 - `host`: 大小为 n 的数组；`host[i]` 表示阶段 i 的主持人。
 - `protocol`: 大小为 n 的数组；`protocol[i]` 表示在阶段 i ($0 < i < n$) 所采用的方式的代码: 0 代表 `IAmYourFriend`, 1 代表 `MyFriendsAreYourFriends`, 而 2 代表 `WeAreYourFriends`。
 - 由于在阶段 0 中没有主持人，因此 `host[0]` 和 `protocol[0]` 是没有被定义的，而在你的程序中也不应访问它们。
 - 这个函数应该返回样本可信度总和的最大值。

子任务

有些子任务只会使用其中部分方式，如下表所示。

子任务	分值	n	可信度	采用的方式
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	全部三种方式
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	只有 <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	只有 <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	只有 <code>IAmYourFriend</code>
5	23	$2 \leq n \leq 1,000$	所有可信度值均为 1	只有 <code>MyFriendsAreYourFriends</code> 和 <code>IAmYourFriend</code> 两种方式
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	全部三种方式

实现细节

你只能提交一个文件，名为 `friend.c`、`friend.cpp` 或 `friend.pas`。在文件中需要实现前

面所述的函数，并遵循如下命名与接口信息。对于C/C++ 程序，你还要加入头文件friend.h。

C/C++ 程序

```
int findSample(int n, int confidence[], int host[], int protocol[]);
```

Pascal 程序

```
function findSample(n: longint, confidence: array of longint, host: array of longint; protocol: array of longint): longint;
```

评测方式

评测系统将会读入如下格式的输入数据：

- 第 1 行: n
- 第 2 行: confidence[0], ..., confidence[n-1]
- 第 3 行: host[1], protocol[1], host[2], protocol[2], ..., host[n-1], protocol[n-1]

评测系统将会输出findSample的返回值。