



Friend 朋友

我們建立了一個由 n 個人組成的社交網絡，這些人的編號為 $0, \dots, n-1$ 。這個網絡中的某對人會成為朋友。如果 X 號人和 Y 號人成為朋友，則 Y 號人同樣地也會和 X 號人成為朋友。

這些人將會在 n 個階段中加入到這個網絡，這些階段也編號為由 0 至 $n-1$ 。第 i 號人會在第 i 個階段被加入。在階段 0 ， 0 號人將會被加入並成為網絡中唯一的人。而在之後的 $n-1$ 個階段中，都有一個人會被主持人加入到網絡中，而這個主持人可以是已經在網絡中的任何一個人。在階段 i 中 ($0 < i < n$)，這個階段的主持人可以從以下三個方法中的其中一個把第 i 號人加入到網絡中：

- *IAmYourFriend* 會使第 i 號人只和主持人成為朋友。
- *MyFriendsAreYourFriends* 會使第 i 號人和主持人這時候的每一個朋友都成為朋友。注意這個方法並不會使第 i 號人和主持人成為朋友。
- *WeAreYourFriends* 會使第 i 號人和主持人成為朋友，同時也會和主持人這時候的每一個朋友都成為朋友。

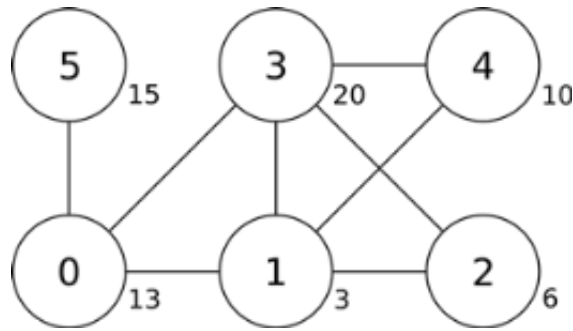
在建立了這個網絡以後，我們想挑選一個調查的樣本，就是說要從網絡中選擇一組人。由於朋友之間通常會有類似的興趣，因此這個樣本中不應該包含任何一對互相成為朋友的人。每個人都會有一個調查信心值，以一個正整數來表示，我們就是要找出一個信心值總和最大的樣本。

例子

階段	主持人	方法	被加入的朋友關係
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)

開始時在網絡中只包含 0 號人。階段 1 的主持人 (0 號人) 用 *IAmYourFriend* 方法去邀請 1 號人，因此他們成為朋友。階段 2 的主持人 (同樣是 0 號人) 用

MyFriendsAreYourFriends 方法去邀請 2 號人，這樣使 1 號人 (主持人的唯一一個朋友) 只和 2 號人成為朋友。階段 3 的主持人 (1 號人) 用 *WeAreYourFriends* 方法加入 3 號人，這樣使 3 號人和 1 號人 (主持人)、 0 號人和 2 號人 (主持人的朋友) 三個都成為朋友。階段 4 和階段 5 都如上述的表格所示。而最後形成的網絡亦如下圖所示，在圖中圓圈內的數值表示他是第幾號人，而在圓圈旁的數值則表示這個人的調查信心值。由 3 號人和 5 號人組成的樣本的調查信心值總和等於 $20 + 15 = 35$ ，這是所有可能的信心值總和中的最大值。



任務

我們給出每個階段的描述和每個人的信心值，請找出一個信心值總和最大的樣本。你只需要編寫一個叫做 `findSample` 的函數。

- `findSample(n, confidence, host, protocol)`
 - `n`: 人的數目。
 - `confidence`: 長度為 n 的陣列; `confidence[i]` 表示第 i 號人的信心值。
 - `host`: 長度為 n 的陣列; `host[i]` 表示階段 i 的主持人。
 - `protocol`: 長度為 n 的陣列; `protocol[i]` 表示在階段 i ($0 < i < n$) 所使用的方法編碼: 0 代表 `IamYourFriend`, 1 代表 `MyFriendsAreYourFriends`, 而 2 代表 `WeAreYourFriends`。
 - 因為在階段 0 中沒有主持人，因此 `host[0]` 和 `protocol[0]` 是沒有被定義的，而在你的程式中你也不應該使用到它們。
 - 這個函數應該返回一個樣本中信心值總和的最大值。

子任務

有些子任務只會使用其中某些方法，如下表所示。

子任務	分數	n	信心值	使用的方法
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	全部三種方法
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	$2 \leq n \leq 1,000$	全部 <code>confidence</code> 的值皆為 1	只有 <code>MyFriendsAreYourFriends</code> 和 <code>IamYourFriend</code> 二種方法
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	全部三種方法

編程詳情

你必須提交唯一一個檔案，其名為 `friend.c`, `friend.cpp` 或 `friend.pas`。這檔案內需

要實現如上面敘述的並有下面特徵的函數。若你是編寫 C/C++ 程式的話, 你亦需要加入標頭檔 friend.h。

C/C++ 程式

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

Pascal 程式

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

樣例評測器

樣例評測器需要讀入以下格式的資料:

- 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]

樣例評測器將會輸出 findSample 的返回值。