



朋友

我們建構一個由 n 個人 (以 p_0, \dots, p_{n-1} 代表這 n 個人) 所組成的社群網路。在這個網路中，總是找到某兩個人是朋友關係。如果 p_x 成爲 p_y 的朋友，那麼 p_y 也會成爲 p_x 的朋友。

這些人經由 n 個階段加入網路，並且依序由 p_0 標示到 p_{n-1} 。 p_i 在階段 i 加入。在階段 0 中， p_0 被加入成爲這個網路裡一開始唯一的人。在接下來 $n-1$ 個階段的每個階段裡，都會有一個人被『接待者』加到網路中。『接待者』可以是這個網路裡既有的任何人。在階段 i ($0 < i < n$)，這個階段的『接待者』可以透過以下任何一種方法將 p_i 加到網路中：

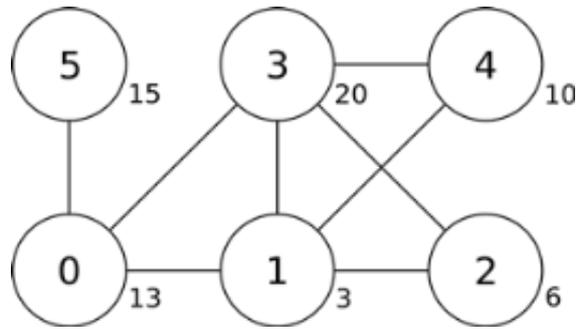
- 『我是你的朋友』 只會讓接待者成爲 p_i 的朋友。
- 『我的朋友就是你的朋友』 會讓接待者在這個時間點的『每個』朋友都成爲 p_i 的朋友。但是要注意的是這個方法『並沒有』讓接待者成爲 p_i 的朋友。
- 『我們都是你的朋友』 會讓接待者以及接待者在這個時間點的『每個』朋友都成爲 p_i 的朋友。

在我們建構完這個網路後，我們想要挑一個『樣本』來進行問卷調查 (也就是說，我們會從這個網路裡挑出一群人來)。由於朋友之間通常有相似的特質，因此我們要求樣本裡沒有任兩個人是朋友的關係。每個人回答問卷時有個『可信度』 (以正整數來表示)，我們想要找出一組可信度總和最大的樣本。

範例

階段	接待者	方法	加入的朋友關係
1	0	我是你的朋友	(1, 0)
2	0	我的朋友就是你的朋友	(2, 1)
3	1	我們都是你的朋友	(3, 1), (3, 0), (3, 2)
4	2	我的朋友就是你的朋友	(4, 1), (4, 3)
5	0	我是你的朋友	(5, 0)

一開始網路裡只有 p_0 。階段 1 的接待者 (p_0) 透過『我是你的朋友』這個方法邀請 p_1 加入，因此這兩人成爲朋友。階段 2 的接待者 (又是 p_0) 透過『我的朋友就是你的朋友』這個方法邀請 p_2 加入，使得 p_1 (這個接待者的唯一朋友) 成爲 p_2 的唯一朋友。階段 3 的接待者 (p_1) 透過『我們都是你的朋友』這個方法將 p_3 加入，使得 p_3 成爲 p_1 (接待者) 以及 p_0 與 p_2 (接待者的朋友) 的朋友。階段 4 與階段 5 的結果也顯示在以上的表格中。最後的網路如下圖所示，其中圓圈裡的數字 i 表示是 p_i ，而圓圈旁的數字則表示他的可信度。包含 p_3 與 p_5 的這個樣本，其可信度的總和爲 $20 + 15 = 35$ ，是所能找到的樣本中，可信度總和最大的。



任務

給定每個階段的描述與每個人的可信度，請找出一個樣本使得其可信度的總和為最大。你只需要實作函式 `findSample`。

- `findSample(n, confidence, host, protocol)`
 - `n`：人的個數。
 - `confidence`：長度為 `n` 的陣列；由 `confidence[i]` 可以得知 `pi` 可信度的數值。
 - `host`：長度為 `n` 的陣列；由 `host[i]` 可以得知階段 `i` 的接待者是誰。
 - `protocol`：長度為 `n` 的陣列；由 `protocol[i]` 可以得知階段 `i` ($0 < i < n$) 所採用的方法為何：0 表示『我是你的朋友』、1 表示『我的朋友就是你的朋友』，而 2 表示『我們都是你的朋友』。
 - 由於階段 0 沒有接待者，因此 `host[0]` 與 `protocol[0]` 都未定義，所以你的程式不應該去存取。
 - 這個函式應該回傳樣本可能存在的最大可信度總和。

子任務

某些子任務只使用了部分的方法，如下表所示：

子任務	得分	n	可信度	採用的方法
1	11	$2 \leq n \leq 10$	$1 \leq \text{可信度} \leq 1,000,000$	所有三種方法
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{可信度} \leq 1,000,000$	只有『我的朋友就是你的朋友』
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{可信度} \leq 1,000,000$	只有『我們都是你的朋友』
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{可信度} \leq 1,000,000$	只有『我是你的朋友』
5	23	$2 \leq n \leq 1,000$	所有可信度的值都是 1	同時有『我的朋友就是你的朋友』跟『我是你的朋友』
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{可信度} \leq 10,000$	所有三種方法

實作細節

你只能送出一個檔案，命名為 `friend.c`、`friend.cpp` 或是 `friend.pas`。這個檔案應該使用以下程式原型來實作上述的副程式。如果你使用 C/C++ 來實作，你也需要引入標頭檔 `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;
```

範例評分程式

範例評分程式讀入以下格式的輸入：

- 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]`

這個範例評分程式將會印出 `findSample` 的回傳值。