



Друг

Ми будуємо соціальну мережу з n людей, пронумерованих $0, \dots, n - 1$. Деякі пари людей в цій мережі будуть друзями. Якщо людина x стає другом людини y , то людина y також стає другом людини x .

Люди добавляються у мережу за n кроків, які також пронумеровані від 0 до $n - 1$. Людина i додається на кроці i . На кроці 0 , додається людина 0 як єдина людина у мережі. На кожному з наступних $n - 1$ кроків наступна людина додається в мережу *господарем*, яким може бути довільна людина, яку вже додано до мережі. На кроці i ($0 < i < n$), господар кроку може додати чергову людину i в мережу за одним з наступних протоколів:

- *IAmYourFriend* робить людину i другом тільки господаря.
- *MyFriendsAreYourFriends* робить людину i другом *кожної* людини, яка є другом господаря у цей момент. Зауважте, що цей протокол *не* робить людину i другом господаря.
- *WeAreYourFriends* робить людину i другом господаря, а також другом *кожної* людини, яка є другом господаря у цей момент.

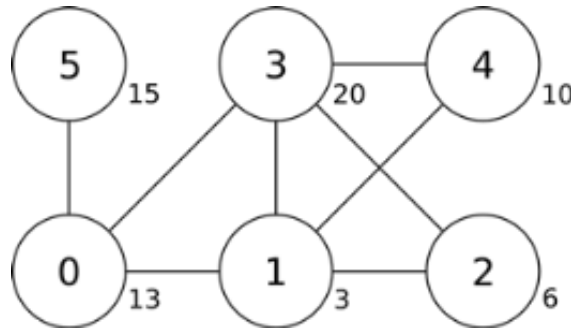
Після того, як ми побудували мережу, ми маємо підібрати *вибірку* для опитування, тобто вибрати групу людей з мережі. Оскільки друзі зазвичай мають спільні інтереси, ця вибірка не повинна містити пари людей, що є друзями. Кожна людина має певний *рівень довіри* у опитуваннях, виражений додатнім цілим числом, і нам треба підібрати вибірку з максимальним загальним рівнем довіри.

Приклад

крок	господар	протокол	додані дружні зв'язки
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)

Спочатку мережа містить тільки людину 0 . Господар кроку 1 (людина 0) запрошує нову людину 1 , використовуючи протокол *IAmYourFriend*, та вони стають друзями. Господар кроку 2 (знову людина 0) запрошує людину 2 , використовуючи *MyFriendsAreYourFriends*, який робить людину 1 (єдиний друг господаря) єдиним другом людини 2 . Господар кроку 3 (людина 1) додає людину 3 , використовуючи *WeAreYourFriends*, що робить людину 3 другом людини 1 (господаря) і людей 0 і 2 (друзі господаря). Кроки 4 і 5 також показано у таблиці вище. Кінцеву мережу показано на наступному рисунку, на якому числа в кружках показують номери людей,

а числа поруч з кружками показують рівень довіри в опитуваннях для цих людей. Вибірка, що складається з людей 3 і 5, має загальний рівень довіри у опитуваннях, що дорівнює $20 + 15 = 35$, що є максимальним можливим загальним рівнем довіри.



Задача

Маючи опис кожного кроку та рівень довіри до кожної людини, знайдіть вибірку з максимальним загальним рівнем довіри. Ви маєте реалізувати функцію `findSample`.

- `findSample(n, confidence, host, protocol)`
 - `n`: кількість людей.
 - `confidence`: масив довжини `n`; `confidence[i]` задає рівень довіри до людини `i`.
 - `host`: масив довжини `n`; `host[i]` задає господаря кроку `i`.
 - `protocol`: масив довжини `n`; `protocol[i]` задає код протоколу, що використовується на кроці `i` ($0 < i < n$): 0 для `IAMYourFriend`, 1 для `MyFriendsAreYourFriends`, та 2 для `WeAreYourFriends`.
 - Оскільки на кроці 0 немає господаря, `host[0]` та `protocol[0]` невизначені і ваша програма не повинна звертатись до них.
 - Функція має повертати максимальний можливий загальний рівень довіри для вибірки.

Підзадачі

Деякі підзадачі використовують не всі протоколи, як показано в таблиці нижче.

підзадача	бали	n	рівень довіри	використаний протокол
1	11	$2 \leq n \leq 10$	$1 \leq \text{рівень довіри} \leq 1,000,000$	Всі три протоколи
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{рівень довіри} \leq 1,000,000$	Тільки <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{рівень довіри} \leq 1,000,000$	Тільки <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{рівень довіри} \leq 1,000,000$	Тільки <code>IAMYourFriend</code>
5	23	$2 \leq n \leq 1,000$	Всі рівні довіри дорівнюють 1	Тільки <code>MyFriendsAreYourFriends</code> та <code>IAMYourFriend</code>
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{рівень довіри} \leq 10,000$	Всі три протоколи

Деталі реалізації

Ви маєте відіслати тільки один файл, що має ім'я `friend.c`, `friend.cpp` або `friend.pas`. Цей файл має реалізовувати підпрограму, що описано вище, використовуючи такі сигнатури. Також підключіть файл заголовків `friend.h` у програму на C/C++.

Програма на C/C++

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

Програма на Pascal

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

Приклад модуля перевірки

Наданий вам модуль перевірки читає вхідні дані у наступному форматі:

- рядок 1: `n`
- рядок 2: `confidence[0], ..., confidence[n-1]`
- рядок 3: `host[1], protocol[1], host[2], protocol[2], ..., host[n-1], protocol[n-1]`

Модуль перевірки надрукує значення, що поверне `findSample`.