



Barát

Egy n tagból álló társaságban a tagokat 0 -tól $n - 1$ -ig sorszámozzuk. Közöttük lehetnek baráti párok. Ha x barátja y -nak, akkor y is barátja x -nek.

A társaságba n lépésben érkeznek a tagok, a lépéseket is a $0..n - 1$ számokkal azonosítjuk. Az i . lépésben az i . tag érkezik. A 0 . lépésben érkező 0 . tag a társaság egyetlen tagja lesz. A további $n - 1$ lépésben az érkező tagot a már ott lévő valamelyik tag lépteti be, akit az i . tag meghívójának hívunk. Az i . lépésben ($0 < i < n$), a meghívó háromféle szabály szerint veheti fel a társaságba az i . tagot:

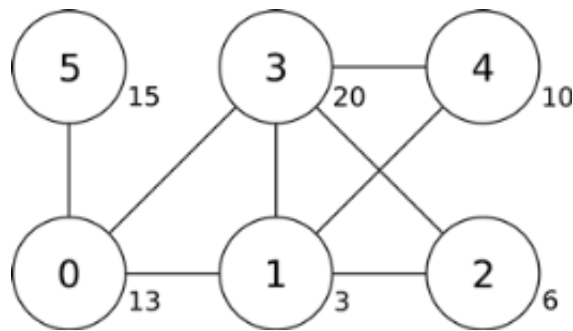
- *IamYourFriend* - az i . tag csak a meghívó barátja lesz.
- *MyFriendsAreYourFriends* - az i . tag a meghívó minden barátjának barátja lesz, de a meghívónak nem.
- *WeAreYourFriends* - az i . tag a meghívónak és a meghívó minden barátjának is barátja lesz.

Kérdőívet szeretnénk kitölteni a társaság kiválasztott tagjaival. Mivel a barátok azonos érdeklődésűek, ezért nem választunk ki barátokat. Minden tagra ismert egy megbízhatósági érték, ami pozitív egész szám. Úgy akarjuk kiválasztani a tagokat, hogy a megbízhatósági értékek összege maximális legyen!

Példa

lépés	meghívó	szabály	az új baráti kapcsolatok
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)

Kezdetben csak a 0 . a társaság tagja. Az első lépésben meghívja az 1 -t, az *IamYourFriend* szabály szerint. A 2 . lépésben szintén a 0 . hívja meg a 2 -at, de a *MyFriendsAreYourFriends* szabály szerint, ezért a 2 -nak a meghívó barátja, azaz az 1 . lesz a barátja. A 3 . lépésben az 1 . hívja meg a 3 -at a *WeAreYourFriends* szabály szerint, azaz a 3 . barátja lesz az 1 . (a meghívó), valamint a 0 . és a 2 . (a meghívó barátai). A 4 . és az 5 . lépés a táblázatban látható. A társaság végső baráti kapcsolatait és a megbízhatóságukat az ábra mutatja. A körökön belül van a tag sorszáma, a körök mellett pedig a megbízhatósága. A 3 . és az 5 . kiválasztása eredményezi a maximális megbízhatósági értékösszeget, ami $20 + 15 = 35$.



Feladat

A lépések és a megbízhatósági értékek ismerete alapján számítsd ki a maximális megbízhatósági értékösszegű csoport megbízhatósági értékösszegét! Ennek megoldására a `findSample` függvényt kell megírnod!

- `findSample(n, confidence, host, protocol)`
 - `n`: a tagok száma.
 - `confidence`: n elemű tömb; `confidence[i]` az i . tag megbízhatósági értéke.
 - `host`: n elemű tömb; `host[i]` az i . lépésben a meghívó sorszáma.
 - `protocol`: n elemű tömb; `protocol[i]` az i . lépésben alkalmazott szabály azonosítója ($0 < i < n$): 0 - IamYourFriend, 1 - MyFriendsAreYourFriends, 2 - WeAreYourFriends.
 - Mivel a 0. lépésben nincs meghívó, ezért `host[0]` és `protocol[0]` értéke definiálatlan, ne hivatkozz rá.
 - A függvény értéke a maximális megbízhatósági értékösszegű csoport megbízhatósági értékösszege legyen!

Részfeladatok

Egyes részfeladatok a meghívási szabályok egy részét használják.

részfeladat	pont	n	megbízhatóság	szabály
1	11	$2 \leq n \leq 10$	$1 \leq \text{megbízhatóság} \leq 1,000,000$	Mindegyik lehet
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{megbízhatóság} \leq 1,000,000$	Csak MyFriendsAreYourFriends
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{megbízhatóság} \leq 1,000,000$	Csak WeAreYourFriends
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{megbízhatóság} \leq 1,000,000$	Csak IamYourFriend
5	23	$2 \leq n \leq 1,000$	minden megbízhatósági érték 1	MyFriendsAreYourFriends és IamYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{megbízhatóság} \leq 10,000$	Mindegyik lehet

Megvalósítás

A friend.c, friend.cpp vagy friend.pas fájlt kell beküldened! Ebben kell megvalósítanod a kért függvényeket! Include-old a friend.h-t!

C/C++ program

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

Pascal program

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

Minta értékelő

A minta értékelő a bemenetet az alábbi formában várja:

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

A minta értékelő a findSample függvény értékét írja ki.