Friend

Construim o rețea de socializare din n persoane, numerotate 0, ... , $n-1$. Unele perechi de persoane din rețea vor deveni prieteni. Dacă persoana x devine prieten al persoanei y, atunci persoana y devine de asemenea prieten al lui x.

Membrii sunt adăugați în rețea în n etape, numerotate de la 0 la $n-1$. Persoana i este adăugată la etapa i. La etapa 0 persoana 0 este adăugată ca unica persoană din rețea. La fiecare din următoarele $n-1$ etape o persoană este adăugată în rețea de o *gazdă*, care trebuie să fie deja un membru al rețelei. La etapa i ($0 < i < n$), gazda acestei etape poate adăuga în rețea persoana cu numărul i folosind unul din următoarele trei protocoale:

- *IAmYourFriend*: persoana i devine prieten doar cu gazda.
- *MyFriendsAreYourFriends*: persoana i devine prieten *cu fiecare* persoană care la acest moment este prieten al gazdei. Atenție, acest protocol *nu* face prieteni persoana i cu gazda.
- *WeAreYourFriends*: persoana i devine prieten atât cu gazda, cât și *cu fiecare* persoană care la acest moment este prieten al gazdei.

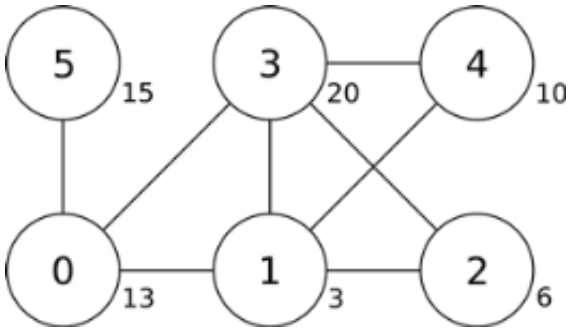
După ce construim rețeaua, dorim să selectăm un *eșantion* pentru un sondaj, ceea ce presupune alegerea unui grup de persoane din rețea. Deoarece prietenii de obicei au interese similare, eșantionul trebuie să nu includă nicio pereche de prieteni. Fiecare persoană are asociată o *încredere* de sondaj, exprimată printr-un număr întreg pozitiv. Vom încerca să determinăm un eșantion de încredere totală maximă (prin încredere totală a unui eșantion se înțelege suma încrederilor asociate presoanelor care îl formează).

Exemplu

etapă	gazdă	protocol	relații de prietenie adăugate
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)

Inițial rețeaua este formată doar din persoana 0. Gazda etapei 1 (persoana 0) invită persoana 1 prin protocolul IAmYourFriend, astfel ei devin prieteni. Gazda etapei 2 (din nou persoana 0) invită persoana 2 prin protocolul MyFriendsAreYourFriends, ceea ce face ca persoana 1 (unicul prieten al gazdei) să devină prieten al persoanei 2 (fără ca persoana 0 să devină prieten cu 2). Gazda etapei 3 (persoana 1) adaugă persoana 3 prin protocolul WeAreYourFriends care face persoana 3 prieten cu persoana 1 (gazda) dar și cu persoanele 0 și 2 (prietenii gazdei). Etapele 4 și 5 sunt de asemenea prezentate în tabelul de mai sus. Rețeaua finală este prezentată în figura următoare, în care numerele înscrise în

cercuri reprezintă indicii membrilor rețelei, iar numerele alăturate cercurilor reprezintă *încrederea* de sondaj asociată persoanelor respective. Eșantionul care constă din persoanele 3 și 5 are încrederea totală egală cu 20 + 15 = 35, care este încrederea totală maximă posibilă.



Cerință

Fiind dată descrierea fiecărei etape și valoarea încrederii asociate fiecărei persoane, găsiți un eșantion cu încredere totală maximă. Trebuie să implementați funcția `findSample`.

- `findSample(n, confidence, host, protocol)`
  - `n`: numărul de persoane.
  - `confidence`: tablou unidimensional de lungime n; `confidence[i]` reprezintă valoarea încrederii asociate persoanei i.
  - `host`: tablou unidimensional de lungime n; `host[i]` reprezintă gazda etapei i.
  - `protocol`: tablou unidimensional de lungime n; `protocol[i]` reprezintă codul protocolului folosit la etapa i ($0 < i < n$): 0 pentru IAmYourFriend, 1 pentru MyFriendsAreYourFriends, și 2 pentru WeAreYourFriends.
  - Deoarece nu există gazdă la etapa 0, `host[0]` și `protocol[0]` sunt nedefinite și nu trebuie să fie accesate de programul vostru.
  - Funcția trebuie să returneze încrederea totală maximă a unui eșantion.

Subprobleme

Unele subprobleme (subtask-uri) vor folosi doar o parte din protocoale, așa cum se vede în tabelul ce urmează.

subtask	puncte	n	încredere	protocoale folosite
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	Toate cele trei protocoale
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Doar MyFriendsAreYourFriends
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Doar WeAreYourFriends
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Doar IAmYourFriend
5	23	$2 \leq n \leq 1,000$	Toate valorile de încredere sunt 1	Doar MyFriendsAreYourFriends și IAmYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	Toate cele trei protocoale

Detalii de implementare

Trebuie să încărcați exact un fișier, numit `friend.c`, `friend.cpp` sau `friend.pas`. Fișierul trebuie să conțină implementarea subprogramului descris mai sus, utilizând următorul antet. De asemenea trebuie să includeți fișierul header `friend.h` pentru implementările C/C++.

pentru programele C/C++

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

pentru programele Pascal

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

Grader-ul de pe computerul vostru

Grader-ul de pe computerul vostru citește datele de intrare în formatul următor:

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

Grader-ul de pe computerul vostru va afișa valoarea returnată de `findSample`.