



## მეგობარი

უნდა ავაგოთ სოციალური ქსელი  $n$  რაოდენობის ადამიანისაგან, რომლებიც გადანორილია 0-დან  $(n - 1)$ -მდე. ქსელში შემაგვალ ადამიანთა ზოგიერთი წყვილი მეგობრები იქნებიან. თუ პერსონა  $x$  ხდება პერსონა  $y$ -ის მეგობარი, მაშინ პერსონა  $y$  ასევე პერსონა  $x$ -ის მეგობარი ხდება.

ადამიანების ქსელში დამატება ხორციელდება  $n$  ეტაპად, რომლებიც ასევე გადანორილია 0-დან  $(n - 1)$ -მდე. პერსონა  $i$  ქსელში  $i$ -ურ ეტაპზე ემატება. 0-ვან ეტაპზე ემატება პერსონა ნომრით 0, როგორც ქსელის ერთადერთი წევრი. ყველა მომდევნო  $n - 1$  რაოდენობის ეტაპზე პერსონას ქსელში ამატებს ე.წ. მასპინძელი (*host*), რომელიც ქსელში უკვე მყოფი ადამიანებიდან ნებისმიერი შეიძლება იყოს.  $i$ -ურ სტადიაზე ( $0 < i < n$ ), ამ სტადიის მასპინძელმა  $i$ -ური პერსონა ქსელში შეიძლება დაამატოს შემდეგი სამი პროტოკოლიდან ერთ-ერთის მიხედვით:

- *IamYourFriend* (მე შენი მეგობარი ვარ), რომელიც  $i$ -ურ პერსონას მხოლოდ მასპინძლის მეგობრად აქცევს.
- *MyFriendsAreYourFriends* (ჩემი მეგობრები შენი მეგობრები არიან), რომელიც  $i$ -ურ პერსონას მასპინძლის *თითოეული* მეგობრის მეგობრად აქცევს. შევნიშნოთ, რომ ეს პროტოკოლი  $i$ -ურ პერსონას მასპინძლის მეგობრად ვერ აქცევს.
- *WeAreYourFriends* (ჩვენ შენი მეგობრები ვართ), რომელიც  $i$ -ურ პერსონას როგორც მასპინძლის, ასევე მისი *თითოეული* მეგობრის მეგობრად აქცევს.

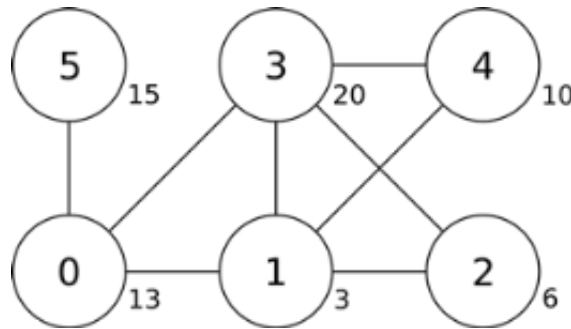
მას შემდეგ, რაც ქსელის აგება დამთავრდება, ჩვენ ქსელში გაერთიანებული ადამიანებისაგან გარკვეული გამოკვლევებისათვის გვსურს *სანიმუშო* ჯგუფის შერჩევა.

რადგანაც მეგობრებს ჩვეულებრივ ერთნაირი ინტერესები აქვთ, ამ ჯგუფში არ უნდა შედიოდეს ერთმანეთის მეგობარ პერსონათა არცერთი წყვილი. ყოველ პერსონას გააჩნია კვლევის თავისი *სანდოობა*, რომელიც მთელი დადებითი რიცხვით გამოისახება და ჩვენი მიზანია ისეთი საკვლევი (სანიმუშო) ჯგუფის შერჩევა, რომელსაც მაქსიმალური ჯამური სანდოობა ექნება.

# მაგალითი

სტადია	მასპინძელი	პროტოკოლი	დამატებულის მეგობრული დამოკიდებულება
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)

თავიდან ქსელში მხოლოდ ერთი ადამიანია ნომრით 0. 1-ლი სტადიის მასპინძელი (პერსონა 0) ამატებს პერსონა 1-ს IamYourFriend პროტოკოლით. ასე რომ, ამ დროიდან ისინი მეგობრები ხდებიან. მე-2 სტადიის მასპინძელი (ასევე პერსონა 0) ამატებს პერსონა 2-ს MyFriendsAreYourFriends პროტოკოლით, რომელიც პერსონა 2-ს მხოლოდ პერსონა 1-ის (რომელიც თავის მხრივ მხოლოდ მასპინძლის მეგობარია) მეგობრად აქცევს. მე-3 სტადიის მასპინძელი (პერსონა 1) ამატებს პერსონა 3-ს WeAreYourFriends პროტოკოლით, რომელიც პერსონა 3-ს პერსონა 1-ის (მასპინძელი) და, ასევე, პერსონა 0-ის და პერსონა 2-ის (მასპინძლის მეგობრები) მეგობრად აქცევს. მე-4 და მე-5 სტადიები ასევე ნაჩვენებია ზემოთ მოცემულ ცხრილში. საბოლოო ქსელი კი ქვემოთაა მოცემული, რომელშიც წრეებში ჩაწერილი რიცხვები ადამიანთა ნომრებია, ხოლო წრეების გვერდით მიწერილი რიცხვები მათი კვლევის სანდოობას გამოსახავს. მე-3 და მე-5 პერსონებისაგან შედგენილ სანიმუშო ჯგუფს აქვს კვლევის ჯამური სანდოობა, რომელიც  $20 + 15 = 35$ -ის ტოლია და იგი კვლევის მაქსიმალურ შესაძლო ჯამურ სანდოობას წარმოადგენს.



## ამოცანა

მოცემულია თითოეული სტადიის აღწერა და თითოეული პერსონის კვლევის სანდოობის მნიშვნელობა. იპოვეთ მაქსიმალური ჯამური სანდოობის მქონე საკვლევი (სანიმუშო) ჯგუფი. თქვენ მხოლოდ უნდა შექმნათ ფუნქცია `findSample`.

- `findSample(n, confidence, host, protocol)`
  - `n`: ადამიანების რაოდენობა.
  - `confidence`:  $n$  სიგრძის მასივი; `confidence[i]` წარმოადგენს  $i$ -ური პერსონის კვლევის სანდოობის მნიშვნელობას.

- `host`:  $n$  სიგრძის მასივი; `host[i]` წარმოადგენს  $i$ -ური სტადიის მასპინძელს.
- `protocol`:  $n$  სიგრძის მასივი; `protocol[i]` წარმოადგენს  $i$ -ურ სტადიაზე გამოყენებული პროტოკოლის კოდს ( $0 < i < n$ ): იგი ტოლია 0-ის `IamYourFriend`-თვის, 1-ის - `MyFriendsAreYourFriends`-თვის და 2-ის - `WeAreYourFriends`-თვის.
- 0-ვან სტადიაზე, სანამ ჯერ კიდევ არ არსებობს მასპინძელი, `host[0]` და `protocol[0]` განსაზღვრული არ არიან და თქვენი პროგრამისათვის მიუწვდომელი უნდა იყვნენ.
- ფუნქციამ უნდა დააბრუნოს სანიმუშო ჯგუფის მაქსიმალური შესაძლებელი ჯამური სანდოობის მნიშვნელობა.

## ქვეამოცანები

ზოგიერთი ქვეამოცანა იყენებს მხოლოდ პროტოკოლთა ქვესიმრავლეს ისე, როგორც ეს ქვემოთ მოცემულ ცხრილშია ნაჩვენები:

ქვეამოცანა	ქულები	$n$	სანდოობა	გამოყენებული პროტოკოლები
1	11	$2 \leq n \leq 10$	$1 \leq \text{სანდოობა} \leq 1,000,000$	სამივე პროტოკოლი
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{სანდოობა} \leq 1,000,000$	მხოლოდ <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{სანდოობა} \leq 1,000,000$	მხოლოდ <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{სანდოობა} \leq 1,000,000$	მხოლოდ <code>IamYourFriend</code>
5	23	$2 \leq n \leq 1,000$	ყველა სანდოობა 1-ის ტოლია	ორივე <code>MyFriendsAreYourFriends</code> და <code>IamYourFriend</code>
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{სანდოობა} \leq 10,000$	სამივე პროტოკოლი

## იმპლემენტაციის დეტალები

თქვენ უნდა წარმოადგინოთ ზუსტად ერთი ფაილი, სახელით `friend.c`, `friend.cpp` ან `friend.pas`. ეს ფაილი იმპლემენტაციას უკეთებს ზემოთ აღწერილ ქვეპროგრამას მითითებული შაბლონების გამოყენებით. თქვენ ასევე გჭირდებათ `friend.h` ფაილი (C/C++)-ში იმპლემენტაციისათვის.

### C/C++ პროგრამა

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

### Pascal პროგრამა

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

## სანიმუშო გრაფერი

სანიმუშო გრაფერი კითხულობს შესატან მონაცემებს შემდეგ ფორმატში:

- 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]

სანიმუშო გრაფერი დაბეჭდავს findSample ფუნქციის მიერ დაბრუნებულ მნიშვნელობას.