



Friend

Vamos construir uma rede social com n pessoas numeradas $0, \dots, n - 1$. Alguns pares de pessoas na rede serão amigas. Se a pessoa x se torna amiga da pessoa y , então a pessoa y também se torna amiga da pessoa x .

As pessoas são adicionadas à rede em n passos, que também são numerados de 0 to $n - 1$. A pessoa i é adicionada no passo i . No passo 0 , a pessoa 0 é adicionada como sendo a única pessoa da rede. Em cada um dos $n - 1$ passos seguintes, uma pessoa é adicionada à rede por um *anfitrião*, que pode ser qualquer pessoa já na rede. No passo i ($0 < i < n$), o anfitrião para aquele passo pode adicionar a nova pessoa i na rede utilizando um dos três protocolos abaixo:

- *IAmYourFriend* torna a pessoa i amiga apenas do anfitrião.
- *MyFriendsAreYourFriends* torna a pessoa i amiga de *cada* pessoa que é amiga do anfitrião neste momento. Note que este protocolo *não* torna a pessoa i amiga do anfitrião.
- *WeAreYourFriends* torna a pessoa i amiga do anfitrião, e também amiga de *cada* pessoa que é amiga do anfitrião neste momento.

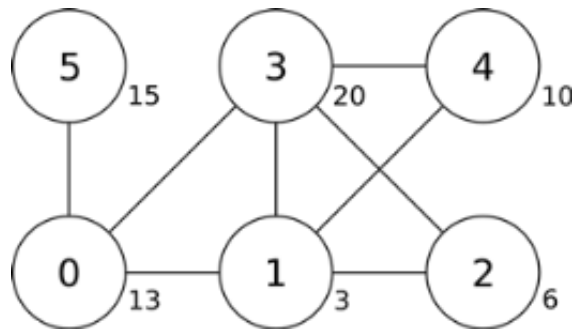
Uma vez construída a rede social, gostaríamos de selecionar uma *amostra* para uma pesquisa, ou seja, escolher um conjunto de pessoas da rede. Uma vez que amigos normalmente têm interesses semelhantes, a amostra não deveria conter nenhum par de amigos. Cada pessoa tem um *nível de confiança*, expresso através de um inteiro positivo, e gostaríamos de determinar uma amostra com o maior total possível de nível de confiança.

Exemplo

passo	anfitrião	protocolo	relações de amizade adicionadas
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)

Inicialmente a rede contém apenas a pessoa 0 . O anfitrião do passo 1 (pessoa 0) convida a nova pessoa 1 utilizando o protocolo *IAmYourFriend*, portanto eles se tornam amigos. O anfitrião do passo 2 (novamente a pessoa 0) convida a pessoa 2 utilizando o protocolo *MyFriendsAreYourFriends*, que torna a pessoa 1 (a única amiga do anfitrião) a única amiga da pessoa 2 . O anfitrião do passo 3 (pessoa 1) adiciona a pessoa 3 utilizando o protocolo *WeAreYourFriends*, que faz a pessoa 3 amiga da pessoa 1 (o anfitrião) e também das pessoas 0 e 2 (as amigas do anfitrião). Os passos 4 e 5 também estão indicados na tabela acima. A rede final está ilustrada na figura abaixo, em que números dentro dos círculos indicam seus números (rótulos), e os números próximos aos círculos indicam o nível de

confiança. A amostra que consiste das pessoas 3 e 5 tem um nível total de confiança igual a $20 + 15 = 35$, que é o maior total possível de nível de confiança.



Tarefa

Dada a descrição de cada passo e o nível de confiança de cada pessoa, determine uma amostra com o maior nível total de confiança. Você precisa implementar apenas a função `findSample`.

- `findSample(n, confidence, host, protocol)`
 - `n`: o número de pessoas.
 - `confidence`: vetor de tamanho `n`; `confidence[i]` fornece o nível de confiança da pessoa `i`.
 - `host`: vetor de tamanho `n`; `host[i]` fornece o anfitrião do passo `i`.
 - `protocol`: vetor de tamanho `n`; `protocol[i]` fornece o código do protocolo utilizado no passo `i` ($0 < i < n$): 0 para `IAmYourFriend`, 1 para `MyFriendsAreYourFriends`, e 2 para `WeAreYourFriends`.
 - Dado que não há anfitrião no passo 0, `host[0]` and `protocol[0]` são indefinidos e não devem ser acessados pelo seu programa.
 - A função deve retornar o maior valor possível do nível total de confiança de uma amostra.

Subtarefas

Algumas subtarefas usam apenas um subconjunto dos protocolos, como indicado na tabela a seguir.

subtarefa	pontos	n	nível de confiança	protocolo utilizado
1	11	$2 \leq n \leq 10$	$1 \leq \text{confiança} \leq 1,000,000$	Todos os três protocolos
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confiança} \leq 1,000,000$	Somente MyFriendsAreYourFriends
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confiança} \leq 1,000,000$	Somente WeAreYourFriends
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confiança} \leq 1,000,000$	Somente IAmYourFriend
5	23	$2 \leq n \leq 1,000$	Todos os valores de confiança são 1	Tanto MyFriendsAreYourFriends como IAmYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	Todos os três protocolos

Detalhes de Implementação

Você tem que submeter precisamente um arquivo, chamado `friend.c`, `friend.cpp` ou `friend.pas`. Este arquivo deve implementar o subprograma descrito acima, usando as seguintes assinaturas. Você deve incluir um arquivo de cabeçalho `friend.h` para implementação em C/C++.

Programas em C/C++

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

Programas em Pascal

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

Avaliador exemplo

O avaliador exemplo lê a entrada no seguinte formato:

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

O avaliador exemplo imprime o valor de retorno de `findSample`.