



Friend (amigo)

Se construye una red social a partir de n personas numeradas $0, \dots, n - 1$. Algunos pares de personas en la red serán amigos. Si la persona x se hace amigo de la persona y , entonces la persona y también se hace amigo de la persona x .

Las personas son añadidas a la red en n etapas, las cuales también son numeradas de 0 a $n - 1$. La persona i es añadida en la etapa i . En la etapa 0 , la persona 0 es añadida como la única persona de la red. En cada una de las siguientes $n - 1$ etapas, la persona es añadida a la red por un *servidor*, que puede ser cualquier persona ya ingresada a la red. Durante la etapa i ($0 < i < n$), el servidor para esa etapa puede añadir la ingresante persona i en la red por uno de los siguientes tres protocolos:

- *IAmYourFriend* hace a la persona i amigo del servidor solamente.
- *MyFriendsAreYourFriends* hace a persona i amigo de *cada* amigo del servidor. Observe que este protocolo *no* hace a la persona i amigo del servidor.
- *WeAreYourFriends* hace a la persona i amigo del servidor, y también amigo de *cada* amigo del servidor.

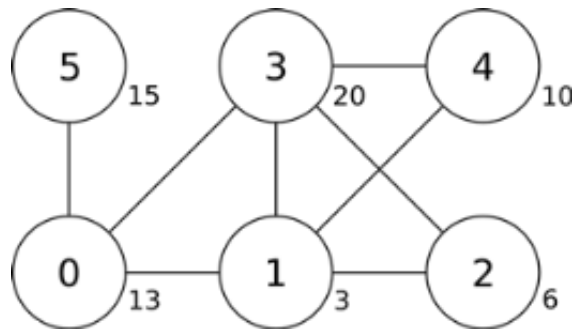
Después de construir la red se desea extraer una *muestra* para una encuesta, eso es, elegir un grupo de personas de la red. Ya que amigos usualmente tienen intereses similares, la muestra no debiera incluir ningún par de personas que sean amigos entre sí. Cada persona tiene un *índice de confianza*, expresado por un entero positivo, y se quiere encontrar una muestra con la máxima confianza total.

Ejemplo

etapa	servidor	protocolo	relaciones amigo añadidas
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 la red contiene sólo la persona 0 . El servidor de etapa 1 (persona 0) invita a la nueva persona 1 a través del protocolo *IAmYourFriend*, por lo que se hacen amigos. El servidor de la etapa 2 (persona 0 nuevamente) invita la persona 2 con *MyFriendsAreYourFriends*, que hace a la persona 1 (el único amigo del servidor) el [único amigo de la persona 2 . El servidor de la etapa 3 (persona 1) añade la persona 3 mediante *WeAreYourFriends*, que hace a la persona 3 amigo de la persona 1 (el servidor) y personas 0 y 2 (los amigos del servidor). Las etapas 4 y 5 son mostradas también en la tabla arriba. La red final es mostrada en la siguiente figura, en la cual los números dentro de los círculos muestran los rótulos de las personas, y los números próximos a los círculos muestran el índice de confianza. La muestra consistiendo de las personas 3 y 5 tiene índice de confianza total igual a $20 +$

15 = 35, el cual es el máximo índice total de confianza posible.



Tarea

Dada la descripción de cada etapa y el valor de confianza de cada persona, encuentre una muestra con la máxima confianza. Solo debes implantar la función `findMuestra`.

- `findMuestra(n, confidence, host, protocol)`
 - `n`: el número de personas.
 - `confidence`: array de largo `n`; `confidence[i]` da el valor confianza de la persona `i`.
 - `host`: array de largo `n`; `host[i]` da el servidor de etapa `i`.
 - `protocol`: array de largo `n`; `protocol[i]` da el código de protocolo usado en la etapa `i` ($0 < i < n$): 0 para `IAmYourFriend`, 1 para `MyFriendsAreYourFriends`, y 2 para `WeAreYourFriends`.
 - Puesto que no hay servidor en etapa 0, `host[0]` y `protocol[0]` están indefinidos y no debieran ser accedidos por tu program.
 - la función debe devolver el máximo a posible valor total de confianza de la muestra.

Subtareas

Algunas subtareas usan sólo un subconjunto de protocolos, tal como se comprueba en la siguiente tabla.

subtarea	puntos	n	confianza	protocolos usados
1	11	$2 \leq n \leq 10$	$1 \leq \text{confianza} \leq 1.000.000$	Todos los tres protocolos
2	8	$2 \leq n \leq 1.000$	$1 \leq \text{confianza} \leq 1.000.000$	Solamente <code>MyFriendsAreYourFriends</code>
3	8	$2 \leq n \leq 1.000$	$1 \leq \text{confianza} \leq 1.000.000$	Solamente <code>WeAreYourFriends</code>
4	19	$2 \leq n \leq 1.000$	$1 \leq \text{confianza} \leq 1.000.000$	Solamente <code>IAmYourFriend</code>
5	23	$2 \leq n \leq 1.000$	Todos los valores de confianza son 1	Tanto <code>MyFriendsAreYourFriends</code> y <code>IAmYourFriend</code>
6	31	$2 \leq n \leq 100.000$	$1 \leq \text{confianza} \leq 10.000$	Los tres protocolos

Detalles de implantación

Debes enviar exactamente un archive, llamado `friend.c`, `friend.cpp` o `friend.pas`. Este archivo deberá implantar el subprograma descrito más arriba, utilizando los siguientes encabezamientos. Necesitas incluir también un archivo cabeza `friend.h` para la implantación en C/C++.

Programación en C/C++

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

Programación en Pascal

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

Sample grader

El sample grader (programa evaluador para prueba local) lee su entrada con el formato siguiente:

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

El sample grader imprimirá el valor de retorno de `findSample`.