Friend

Construimos una red social de n personas numeradas 0,…,$n-1$. Algunos pares de personas en la red serán amigos. Si la persona x se vuelve amiga de la persona y, entonces también la persona y se vuelve amiga de la persona x.

Se añade gente a la red en n etapas, las cuales también están numeradas de 0 a $n-1$. La persona i se añade en la etapa i. En la etapa 0, se añade la persona 0 como la única persona en la red. En cada una de las siguientes $n-1$ etapas, se añade una persona a la red por un encargado, el cual puede ser cualquier persona que ya esté en la red. En la etapa i ($0 < i < n$), el encargado de esa etapa puede añadir a la person i entrante en la red a través de uno de los siguientes tres protocolos:

- *IamYourFriend* hace que la persona i se vuelva amiga del encargado.
- *MyFriendsAreYourFriends* hace que la persona i sea amiga de *cada* amiga del huésped. Note que este protocolo no hace que i sea amiga del encargado.
- *WeAreYourFriends* hace que la persona i sea amiga del encargado y también que sea amiga de *cada* amiga del encargado.

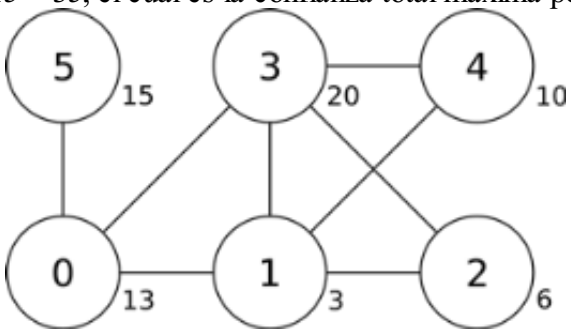
Después de construir la red quisiéramos elegir una *muestra* para una encuesta, esto es, elegir un grupo de gente de la red. Como los amigos tienen generalmente intereses en común, la muestra no debería incluir ningún par de personas que sean amigas, la una con la otra. Cada persona tiene una *confianza* para encuestas, expresada como un entero positivo y quisiéramos encontrar una muestra con la confianza total máxima posible.

Ejemplo

etapa	encargado	protocolo	relaciones de amistad 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 únicamente a la persona 0. El encargado de la etapa 1 (persona 0) invita a la persona 1 a través del protocolo IamYourFriend, por lo tanto se vuelven amigas. El encargado de la etapa 2 (nuevamente la persona 0) invita a la persona 2 con MyFriendsAreYourFriends, lo cual hace que la persona 1 (la única amiga del encargado) sea la única amiga de la persona 3. La persona encargada de la etapa 3 (persona 1) añade a la persona 3 a través de WeAreYourFriends, lo cual hace que la persona 3 una amiga de la persona 1 (la encargada) y de las personas 0 y 2 (las amigas de la encargada). Las etapas 4 y 5 también se muestran en la tabla anterior. La red final se muestra en la siguiente figura, en la cual los números dentro de los círculos muestran a las personas y los números

cerca de los círculos muestran la confianza. La muestra consistente de las personas 3 y 5 tiene la confianza total igual a 20 + 15 = 35, el cual es la confianza total máxima posible.



Tarea

Dada la descripción de cada etapa y la confianza de cada persona, encontrar una muestra con la confianza máxima total. Usted únicamente necesita implementar la función `findSample`.

- `findSample(n, confidence, host, protocol)`
  - `n`: el número de personas.
  - `confidence`: arreglo de tamaño n; `confidence[i]` da la confianza de la persona i.
  - `host`: arreglo de longitud n; `host[i]` da el encargado de la etpa i.
  - `protocol`: arreglo de longitud 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.
  - Como no hay encargado en la etapa 0, `host[0]` y `protocol[0]` están indefinidos y no deberìan ser accesados por su programa.
  - La función debe devolver la confianza total máxima posible para una muestra.

Subtareas

Algunas subtareas usan únicamente un subconjunto de protocolos, como se muestra en la siguiente tabla.

subtarea	puntos	n	confianza	protocolos usados
1	11	$2 \leq n \leq 10$	$1 \leq$ confianza $\leq 1,000,000$	Todos los tres protocolos
2	8	$2 \leq n \leq 1,000$	$1 \leq$ confianza $\leq 1,000,000$	Solo MyFriendsAreYourFriends
3	8	$2 \leq n \leq 1,000$	$1 \leq$ confianza $\leq 1,000,000$	Solo WeAreYourFriends
4	19	$2 \leq n \leq 1,000$	$1 \leq$ confiancia $\leq 1,000,000$	Solo IamYourFriend
5	23	$2 \leq n \leq 1,000$	Todas las confianzas 1	Ambos MyFriendsAreYourFriends y IamYourFriend
6	31	$2 \leq n \leq 100,000$	$1 \leq$ confianza $\leq 10,000$	Todos los tres protocolos

Detalles de implementación

Usted tiene que enviar exactamente un archivo, llamado `friend.c`, `friend.cpp` o `friend.pas`. Este archivo debe implementar el subprograma descrito anteriormente, usando los siguientes prototipos. Usted también debe incluir un archivo de encabezado `friend.h` para implementación en C/C++.

Programas C/C++

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

Programas Pascal

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

Calificador ejemplo

El calificador ejemplo lee la entrada en el siguiente formato:

- 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 calificador ejemplo imprimirá el valor devuelto de `findSample`.