Friend

Kita membangun sebuah jaringan sosial yang terdiri atas n orang yang dinomori 0, ... , $n-1$. Beberapa pasang orang di jaringan tersebut akan menjadi teman. Jika orang x menjadi teman dari orang y, maka orang y juga menjadi teman dari orang x.

Orang-orang tersebut ditambahkan ke jaringan dalam n tahapan, dengan tahapan tersebut juga dinomori 0 sampai dengan $n-1$. Orang ke-i ditambahkan pada tahap ke- i. Pada tahap ke-0, orang ke-0 ditambahkan sebagai satu-satunya orang dalam jaringan. Pada setiap $n-1$ tahap berikutnya, seseorang ditambahkan ke jaringan oleh seorang *host*, dengan *host* adalah salah satu dari sebarang orang yang sudah berada dalam jaringan. Pada tahap ke-i ($0 < i < n$), host dari tahap tersebut dapat menambahkan orang ke-i yang akan datang ke dalam jaringan dengan salah satu dari tiga protokol berikut:

- *IamYourFriend* menjadikan orang ke-i hanya sebagai teman dari *host* saja.
- *MyFriendsAreYourFriends* menjadikan orang ke-i sebagai teman dari *tiap* teman *host* saat itu. Perlu diingat, protokol ini *tidak* menjadikan orang ke-i sebagai teman dari *host*.
- *WeAreYourFriends* menjadikan orang ke-i sebagai teman dari *host*, dan menjadikan juga orang ke-i sebagai teman dari *tiap* teman dari *host*.

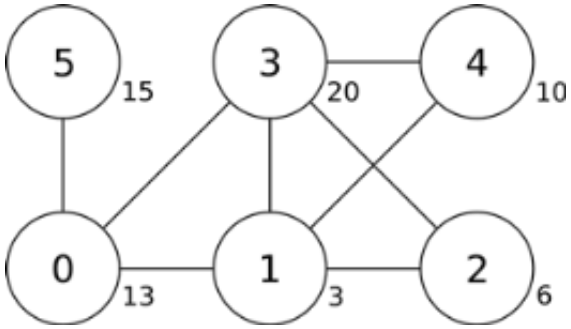
Setelah kita membangun jaringan, kita ingin mengambil sebuah *sample* untuk sebuah survey, yakni memilih sekelompok orang di dalam jaringan. Karena orang yang berteman biasanya mempunyai minat yang sama, sampel tidak boleh mengikutkan pasangan orang yang sudah berteman satu sama lain. Setiap orang akan mempunyai nilai survey *confidence*, yang dinyatakan dengan bilangan integer positif, dan kita ingin untuk mendapatkan sampel dengan jumlah total *confidence* yang maksimum.

Example

tahap	host	protokol	relasi teman yang ditambahkan
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)

Mula-mula, jaringan hanya berisi orang 0. *Host* pada tahap 1 (orang 0) mengundang seseorang yang baru yaitu orang 1 dengan protokol IamYourFriend, sehingga orang 1 menjadi teman orang 0. *Host* pada tahap 2 (tetap orang 0) mengundang orang 2 dengan protokol MyFriendsAreYourFriends, yang menjadikan orang 1 (satu-satunya teman dari *host*) menjadi teman dari orang 2. *Host* pada tahap 3 (orang 1) menambahkan orang 3 dengan protokol WeAreYourFriends, yang menjadikan orang 3 sebagai teman dari orang 1 (*host*) sekaligus orang 0 dan 2 (teman dari *host*). Tahap 4 dan 5 juga

ditunjukkan dalam tabel di atas. Keadaan akhir dari jaringan ditunjukkan dalam gambar berikut, dengan nomor di dalam lingkaran menunjukkan label seseorang, dan angka di samping lingkaran menunjukkan nilai survey *confidence*. Sampel yang terdiri dari orang 3 dan 5 mempunyai jumlah total survey *confidence* sama dengan 20 + 15 = 35, yang merupakan total *confidence* maksimal yang mungkin.



Task

Berdasarkan deskripsi dari tiap tahapan dan nilai *confidence* dari tiap orang, tentukan sebuah sample dengan total *confidence* maksimum. Anda diminta untuk mengimplementasikan sebuah fungsi `findSample`.

- `findSample(n, confidence, host, protocol)`
  - `n`: the number of people.
  - `confidence`: array of length n; `confidence[i]` gives the confidence value of person i.
  - `host`: array of length n; `host[i]` gives the host of stage i.
  - `protocol`: array of length n; `protocol[i]` gives the protocol code used in stage i ($0 < i < n$): 0 for IamYourFriend, 1 for MyFriendsAreYourFriends, and 2 for WeAreYourFriends.
  - Since there is no host in stage 0, `host[0]` and `protocol[0]` are undefined and should not be accessed by your program.
  - The function should return the maximum possible total confidence of a sample.

Subtasks

Some subtasks use only a subset of protocols, as shown in the following table.

subtask	points	n	confidence	protocols used
1	11	$2 \le n \le 10$	$1 \le \text{confidence} \le 1,000,000$	All three protocols
2	8	$2 \le n \le 1,000$	$1 \le \text{confidence} \le 1,000,000$	Only MyFriendsAreYourFriends
3	8	$2 \le n \le 1,000$	$1 \le \text{confidence} \le 1,000,000$	Only WeAreYourFriends
4	19	$2 \le n \le 1,000$	$1 \le \text{confidence} \le 1,000,000$	Only IamYourFriend
5	23	$2 \le n \le 1,000$	All confidence values are 1	Both MyFriendsAreYourFriends and IamYourFriend

subtask	points	n	confidence	protocols used
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	All three protocols

Implementation details

You have to submit exactly one file, called `friend.c`, `friend.cpp` or `friend.pas`. This file should implement the subprogram described above, using the following signatures. You also need to include a header file `friend.h` for C/C++ implementation.

C/C++ program

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

Pascal programs

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

Sample grader

The sample grader reads the input in the following format:

- 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]`

The sample grader will print the return value of `findSample`.