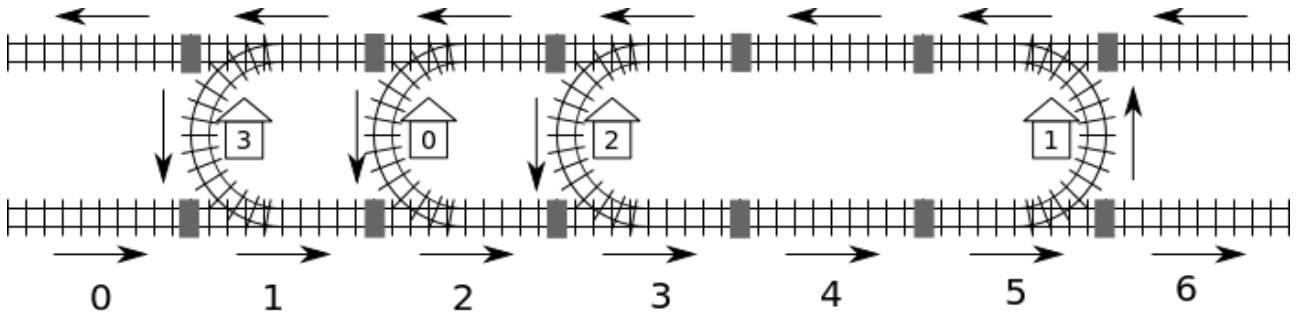




Demiryolu

Tayvan'da adanın batı ve doğu kıyılarını birbirine bağlayan uzun bir demiryolu hattı vardır. Bu demiryolu hattı m bloktan meydana gelmektedir. Ardışık bloklar batı kıyısından doğu kıyısına 0'dan $m - 1$ 'e numaralandırılmıştır. Her bir blokta, kuzeyde doğudan batıya doğru bir tek yönlü hat ve güneyde de batıdan doğuya doğru bir tek yönlü hat bulunmaktadır. Bazı bloklarda kuzey ve güney hatları arasında tercihen bir tren istasyonu bulunabilir.

Üç tip blok vardır. C tipindeki bloklarda sadece kuzeyden girilip güneyden çıkılabilen bir istasyon bulunmaktadır. D tipindeki bloklarda da sadece güneyden girilip kuzeyden çıkılabilen bir istasyon bulunmaktadır. Boş tipindeki bloklarda hiç istasyon bulunmamaktadır. Örnek vermek gerekirse, aşağıdaki şekilde 0, 4 ve 6. blokların tipi boş, 1, 2 ve 3. blokların tipi C ve blok 5'in tipi D 'dir. Bloklar birbirlerine bağlantılarla bağlanmışlardır. Aşağıdaki şekilde bağlantılar gri dikdörtgenlerle gösterilmiştir.



Demiryolu sisteminde 0'dan $n - 1$ 'e kadar numaralandırılmış n tane istasyon vardır. Herhangi bir istasyondan başka bir istasyona rayları takip ederek gitmenin mümkün olduğunu varsayınız. Örnek vermek gerekirse, istasyon 0'dan istasyon 2'ye, 2. bloktan başlayıp güneydeki raylarla 3. ve 4. blokları geçip sonra 1. istasyon üzerinden 5. bloktan geçtikten sonra 4. bloktan geçip, son olarak 3. bloktaki 2. istasyona gelerek ulaşmak mümkündür.

İki istasyonu bağlayan birden fazla rota olabileceğinden, bir istasyondan başka bir istasyona olan mesafe en az bağlantı içeren rotadaki bağlantı sayısı olarak tanımlanmıştır. Örnek olarak, 0. istasyondan 2. istasyona olan en kısa rota 2-3-4-5-4-3 bloklarından 5 bağlantı kullanılarak oluşturulmuş için mesafe 5'tir.

Demiryolu sistemini bir bilgisayar sistemi idare etmektedir. Ne yazık ki bir elektrik kesintisi sonucu, bilgisayar artık istasyonların nerede olduklarını ve buldukları blok tipi bilgilerini kaybetmiştir. Bilgisayarın tek bildiği, 0. istasyonun kaçınıcı blokta olduğu ve o bloğun her zaman C tipinde olduğudur. Bilgisayar ayrıca bir istasyondan başka bir istasyona olan mesafeyi sorgulayabilir. Örnek olarak, '0. istasyondan 2. istasyona olan mesafe nedir?' sorgusunu yapabilir ve cevap olarak 5'i bulabilir.

Görev

`findLocation` adında, herhangi bir istasyonun kaçınıcı blokta olduğunu ve bulunduğu bloğun tipini bulan bir fonksiyon yazmanız gerekmektedir.

- `findLocation(n, first, location, stype)`
 - `n`: istasyonların sayısı.
 - `first`: 0. istasyonun bulunduğu blok numarası.
 - `location`: n uzunluğunda bir dizi; i . istasyonun blok numarasını `location[i]`'ye yazmalısınız.
 - `stype`: n uzunluğunda bir dizi; i . istasyonun blok tipini `stype[i]`'ye yazmalısınız: C tipi için 1 ve D tipi için 2 yazmalısınız.

İstasyonların yerlerini ve tiplerini bulmada yardımcı olmak için `getDistance` adlı fonksiyonu çağırabilirsiniz.

- `getDistance(i, j)` i . istasyondan j . istasyona olan mesafeyi döner. `getDistance(i, i)` 0 döner. `getDistance(i, j)` eğer i ya da j $0 \leq i, j \leq n - 1$ sınırları içinde değilse -1 döner.

Alt görevler

Bütün alt görevlerde blok sayısı m 1,000,000'a eşit ya da daha küçüktür. Bazı alt görevlerde `getDistance` fonksiyonunu çağırma sayısı sınırlanmıştır. Sınır alt görevden alt göreve değişebilir. Programınız fonksiyon çağırma sınırını aşarsa 'yanlış cevap' olarak değerlendirilecektir.

alt görev	puan	n	getDistance sayısı	not
1	8	$1 \leq n \leq 100$	sınırsız	istasyon 0 dışındaki bütün istasyonlar D tipindedir.
2	22	$1 \leq n \leq 100$	sınırsız	0. istasyonun doğusundaki bütün istasyonlar D tipi bloklardadır ve 0. istasyonun batısındaki bütün istasyonlar C tipi bloklardadır.
3	26	$1 \leq n \leq 5,000$	$n(n - 1)/2$	ek bir sınır yoktur
4	44	$1 \leq n \leq 5,000$	$3(n - 1)$	ek bir sınır yoktur

Gerçekleştirim detayları

İsmi `rail.c`, `rail.cpp` ya da `rail.pas` olan bir dosya göndermelisiniz. Bu dosya `findLocation` fonksiyonunu aşağıda verilen yapıda gerçekleştirir. C/C++ programı için `rail.h` header dosyasını eklemelisiniz.

C/C++ programları

```
void findLocation(int n, int first, int location[], int stype[]);
```

Pascal programları

```
procedure findLocation(n, first : longint; var location,
```

```
styp e : array of longint);
```

getDistance fonksiyonunun yapısı aşağıdaki gibidir.

C/C++ programları

```
int getDistance(int i, int j);
```

Pascal programları

```
function getDistance(i, j: longint): longint;
```

Örnek grader

Örnek grader girdi'yi aşağıdaki formatta okur:

- satır 1: altgörev numarası
- satır 2: n
- satır $3 + i$, ($0 \leq i \leq n - 1$): $styp e[i]$ (C tipi için 1 ve D tipi için 2), $location[i]$.

Örnek grader, findLocation fonksiyonu döndüğünde eğer programınız tarafından hesaplanan $location[0] .. location[n-1]$ ve $styp e[0] ... styp e[n-1]$ dizileri girdiyle eşleşirse Correct yazar, eğer eşleşmezse Incorrect yazar.