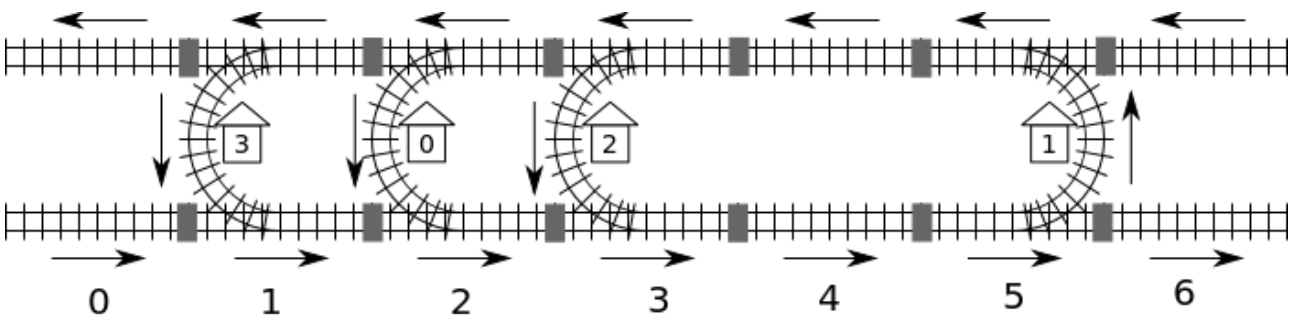




## Rail 鐵路

台灣有一個很大的鐵路系統連接著島的東岸及西岸。這個鐵路系統包含有  $m$  個模塊。這些相連的模塊以數字  $0, \dots, m-1$  加以編號之，編號是由西岸開始。每一個模塊都包含一段在北面單行向西的路軌及一段在南面單行向東的路軌。有部份模塊亦有可能會有一個位於這兩段路軌之間的車站。

現共有三種不同類型的模塊。其中  $C$  類型的模塊將包含一個車站，且這車站必須由北面的路軌進入且由南面的路軌離開。而一個  $D$  類型的模塊亦包含一個車站，且該車站必須由南面的路軌進入且由北面的路軌離開。而一個空 (empty) 類型的模塊則是一個不包含車站的模塊。例如下圖所示，模塊 0, 4 及 6 均是空類型模塊，模塊 1, 2 及 3 是  $C$  類型的模塊。相鄰的模塊之間以一個連接器互相連接著，在圖中所示的灰色小長方塊就是連接器。



在這個鐵路系統中共有  $n$  個車站，它們分別以數字 0 至  $n-1$  加以編號之。這裡我們假設我們可以由一任何個車站出發，沿著鐵路的方向走是可以走到任何其他一個車站。例如我們可以由車站 0 去到車站 2 所經路線是由模塊 2 開始，經模塊 3 及模塊 4 的南面路軌，然後在模塊 5 通過車站 1，再經模塊 4 的北面路軌，最後到達模塊 3 並由北面路軌進入車站 2。

由於由一個車站到另一個車站有很多不同的可能路線，我們定義由一個車站到另一個車站的最短距離為路線中所經的連接器的最少數目。例如由車站 0 到車站 2 所經的模塊是 2-3-4-5-4-3 所經的連接器數目為 5，因此它們的路離是 5。

這鐵路資料原本是由一個電腦系統來管理。不幸地一次大停電後電腦失去了各車站所在的模塊編號及其相應的類型。電腦系統目前唯一知道的是車站 0 所在的模塊編號，並知道該模塊的類型是  $C$  類型。幸運的是你可以向電腦查詢任何兩個車站之間的路離。例如你可以問電腦 '車站 0 與車站 2 的距離'，它會給你答案 5。

## 任務

你需要編寫一個函數 `findLocation` 用以決定每一個車站所在的模塊編號及其相應的類型。

- `findLocation(n, first, location, stype)`

- `n`: 車站的數目

- first: 車站 0 所在模塊的編號
- location: 是一個大小為  $n$  的陣列; 你應該將車站  $i$  所在的模塊編號放在 `location[i]` 內
- stype: 是一個大小為  $n$  的陣列; 你應該將車站  $i$  所在的模塊的類型資料存放在 `stype[i]` 內: 其中 1 代表 C 類型而 2 則代表 D 類型

你可以呼叫一名為 `getDistance` 來幫助你找出一個車站所在模塊的編號及類型。

- `getDistance(i, j)` 將返回車站  $i$  到車站  $j$  的距離。若呼叫 `getDistance(i, i)` 將得返回值 0。若 `getDistance(i, j)` 將返回 -1 若  $i$  或  $j$  在  $0 \leq i, j \leq n - 1$  範圍以外。

## 子任務

在所有的子任務中, 模塊的數目  $m$  是不會大於 1,000,000 的。在部份子任務中, 容許呼叫 `getDistance` 的次數是有限的, 且該限制將會是因子任務而異。你的程式將得到 'wrong answer' 若呼叫的次數大於容許的上限。

子任務	得分	$n$	getDistance 呼叫次數	備註
1	8	$1 \leq n \leq 100$	沒限制	除車站 0 外其他所有車站所在的模塊均是 D 類型。
2	22	$1 \leq n \leq 100$	沒限制	所有在車站 0 東邊的車站所在的模塊均是 D 類型, 而所有在車站 0 西邊的車站所在的模塊均是 C 類型的。
3	26	$1 \leq n \leq 5,000$	$n(n-1)/2$	沒有其他限制
4	44	$1 \leq n \leq 5,000$	$3(n-1)$	沒有其他限制

## 實現詳情

你必須提交唯一一個檔案, 其名為 `rail.c`, `rail.cpp` 或 `rail.pas`。這檔案內需要實現如上面敘述所講的 `findLocation` 並包括下面的特徵。若你是編寫 C/C++ 程式的話, 你亦需要加入標頭檔 `rail.h`。

### C/C++ 程式

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

### Pascal 程式

```
procedure findLocation(n, first : longint; var location,
  stype : array of longint);
```

`getDistance` 的特徵如下:

### C/C++ 程式

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

## Pascal 程式

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

## Sample grader 樣例評測器

樣例評測器需要讀入以下格式的資料:

- 行 1: 子程式編號
- 行 2:  $n$
- 行  $3 + i, (0 \leq i \leq n - 1)$ :  $stype[i]$  (1 代表 C 類型 2 代表 D 類型),  $location[i]$ .

樣例評測器將輸出 Correct 若你程式在 `findLocation` 完成並返回時所找到的  $location[0] \dots location[n-1]$  及  $stype[0] \dots stype[n-1]$  和輸入資料相同, 否則它會輸出 Incorrect。