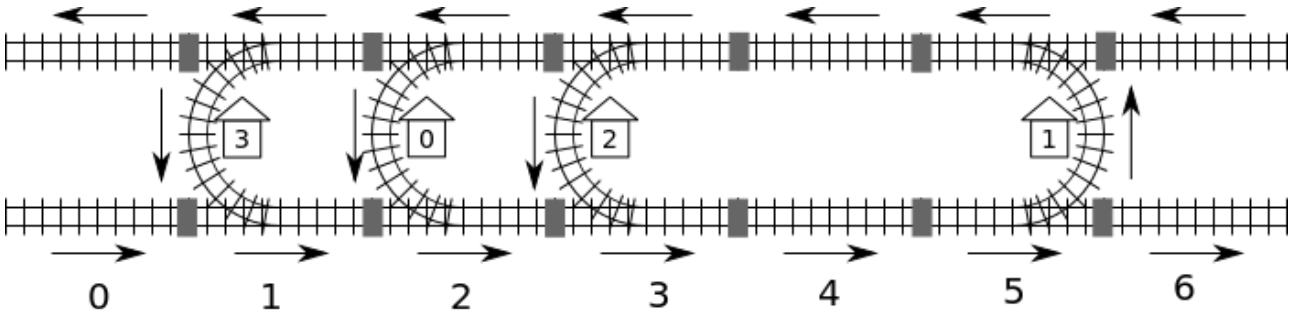
台灣橫貫鐵路

在百年後的二十二世紀，台灣建立了一組貫穿中央山脈的橫貫鐵路系統連接台灣的西岸與東岸。這組鐵路共分成 m 個區域，從西到東命名爲區域0、區域1、區域2、...、到區域$m-1$。每個區域的北邊是一個向西的單向鐵路，而每個區域的南邊是一個向東的單向鐵路。而在每個區域內部可能設有一個縱向單軌車站連接南北邊的橫向鐵路，也可能在該區域中沒有設置車站。

這些區域可以分成三種型態。*型態C* 是指該區域中設有一個車站並配備一個向南的單軌鐵路。*型態D* 是指該區域中設有一個車站並配備一個向北的單軌鐵路。*型態E* (empty) 是指該區域中沒有設置車站。如圖中，區域0、區域4、區域6是屬於 *型態E*，區域1、區域2、區域3是屬於 *型態C*， 區域5 是屬於 *型態D* 。如圖中，所有的區域是水平地排列在一起。各區域北邊的鐵路使用 *連接器*(深色矩形方塊)形成一個向西的單向鐵路。各區域南邊的鐵路使用 *連接器*形成一個向東的單向鐵路。



這個鐵路系統有 n 個車站，編號爲 車站0、車站1、...、車站$n-1$。已知任一個車站可以透過這個單軌鐵路系統到達任一個車站。舉例來說，我們可以從車站0到達車站2 ，只要我們從車站0所在的區域2往南、然後往東通過區域3、區域4，然後透過區域5的車站1往北，向西通過區域4，來到區域3到達車站2。

車站A 到 車站B 走法的長度就定義爲該走法通過的 *連接器*個數。而 車站A 到 車站B 的走法可能不只一條，所以車站A 到 車站B 的距離就定義爲 *最短走法的長度*。例如車站0到車站2的最短走法就是 區域2-區域3-區域4-區域5-區域4-區域3，而該走法通過 5 個 *連接器*，所以其距離爲5。

二十二世紀台灣橫貫鐵路是由一個超級電腦管理，但是不幸的發生了台灣有史以來的最大的地震造成全台大停電，復電之後，這台超級電腦儲存各車站所在區域位置與其型態(C或D)的資料受損，唯一知道的是車站0的型態爲C，以及車站0所在區域的編號。但是幸運的是，儲存各車站間最短距離的大表仍然存在。例如 你可以詢問"車站0到車站2的距離"， 該超級電腦將會回答你"距離爲5"。

任務

現在請您撰寫 findLocation 函數來協助該超級電腦恢復各車站的型態及其所在區域。

- findLocation(n, first,location, stype)
  - n: 車站總數(總共有n個車站)。
  - first: 車站0所在區域編號(車站0在區域$first$中)。
  - location: 個數爲n的陣列，你應該將車站i所在的區域編號填入 location[i]之中。
  - stype: 個數爲n的陣列，你應該將車站i的型態資訊填入stype[i]中，但是注意*型態C*請填1，*型態D*請填2。

你可以呼叫 getDistance 來幫助你尋找車站所在區域及型態。

- getDistance(i,j) 會回傳車站i到車站j的距離(最短路徑長度)。
  如果你詢問 車站i到車站i的距離，那麼 getDistance(i,i) 將會回傳0。
  如果你詢問 不存在的車站編號，例如 i 或 j 不在0到$n-1$之間，那麼 getDistance(i,j)將會回傳-1。

子任務

在所有的子任務中，區域的個數 m 將不會超過 1,000,000。而在某些子任務中，呼叫 getDistance 的次數將會受到限制，限制的次數將會隨子任務而不同。如果你呼叫的次數超過要求，繳交的檔案將會認定爲錯誤的答案(wrong answer)。

子任務	分數	n	getDistance 呼叫次數	說明
1	8	$1 \le n \le 100$	不限制	除了車站0以外，其餘車站都是*型態D*。
2	22	$1 \le n \le 100$	不限制	車站0以東的車站都是*型態D*，車站0以西的車站都是*型態C*。
3	26	$1 \le n \le 5,000$	$n(n-1)/2$	無限制
4	44	$1 \le n \le 5,000$	$3(n-1)$	無限制

撰寫注意事項

你必須繳交一個檔案，檔名依照你使用的語言可以是 rail.c, rail.cpp 或 rail.pas。
如果你使用C/C++語言，你必須引用rail.h。在繳交檔中，你必須使用下列的函式原型來撰寫 findLocation 函式。

C/C++ program

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

Pascal program

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

```
  stype : array of longint);
```

getDistance 函式的原型如下。

C/C++ program

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

Pascal program

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

範例評分函式

範例評分函式將會依照下列的格式讀取輸入資料：

- 第 1 行： 子任務的編號
- 第 2 行：n
- 第 $3+i$ 行, ($0 \leq i \leq n-1$)： stype[i] (1 爲*型態C*，2爲*型態D*), location[i]。

如果您的函式findLocation計算的location[0] ... location[n-1] 和 stype[0] ... stype[n-1]與答案相符，評分函式將會列印 "Correct"，否則將會列印 "Incorrect"。