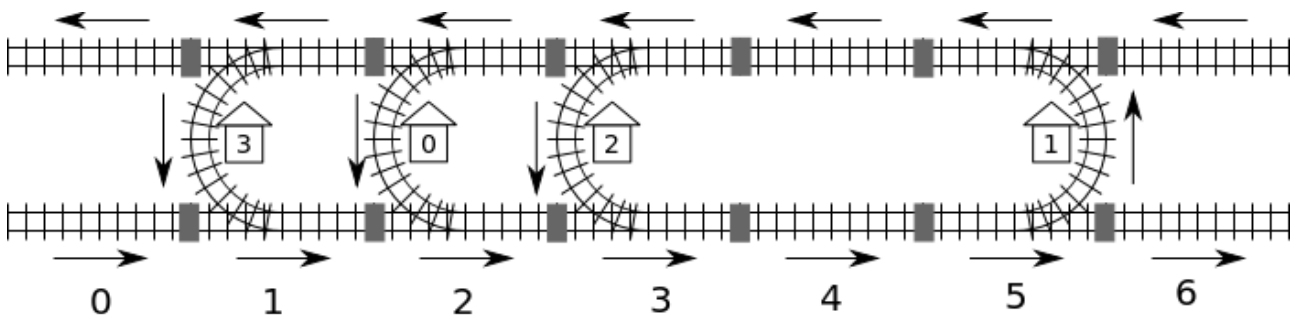




Đường tàu

Đài Loan có một hệ thống đường tàu lớn nối hai bờ đông và tây của hòn đảo. Hệ thống đường tàu có m khối. Các khối nằm liên tiếp cạnh nhau được đánh số thứ tự $0, \dots, m - 1$, bắt đầu từ đầu bờ phía tây. Mỗi một khối gồm hai tuyến đường một chiều: tuyến đường ở phía bắc đi về hướng tây, tuyến đường ở phía nam đi về hướng đông; trong mỗi khối còn có thể có một ga tàu nằm giữa hai tuyến đường này.

Các khối được chia thành ba loại. Khối loại *C* chứa một ga tàu mà bạn phải đi vào từ tuyến đường nằm ở phía bắc và thoát ra tuyến đường nằm ở phía nam, khối loại *D* chứa một ga tàu mà bạn phải đi vào từ tuyến đường nằm ở phía nam và thoát ra tuyến đường nằm ở phía bắc, khối loại *trống* không chứa ga tàu nào. Ví dụ, trong hình vẽ dưới đây, khối 0, 4 và 6 thuộc loại trống, khối 1, 2 và 3 thuộc loại *C*, và khối 5 thuộc loại *D*. Các khối nối với nhau theo chiều ngang. Các tuyến tương ứng của hai khối liền kề được thông với nhau bởi các *điểm nối*, được minh họa bởi các hình chữ nhật tô mờ trong hình vẽ dưới đây.



Hệ thống đường tàu có n ga tàu được đánh số từ 0 đến $n - 1$. Có thể giả thiết rằng *ta có thể đi từ bất kỳ một ga tàu này đến một ga tàu khác* bằng cách đi theo tuyến đường. Ví dụ ta có thể đi từ ga tàu 0 đến ga tàu 2 bằng cách xuất phát từ khối 2, rồi đi qua khối 3 và 4 sử dụng tuyến đường nằm ở phía nam, và đi qua khối 5 qua ga tàu 1, sau đó đi qua khối 4 sử dụng tuyến đường nằm ở phía bắc, và cuối cùng đến ga tàu 2 nằm ở khối 3.

Vì có thể có nhiều đường đi giữa hai ga tàu, khoảng cách từ một ga tàu đến một ga tàu khác được định nghĩa là số lượng *nhỏ nhất* các điểm nối mà đường đi phải đi qua. Ví dụ, đường đi ngắn nhất từ ga tàu 0 đến ga tàu 2 đi qua khối 2-3-4-5-4-3 và đi qua 5 điểm nối, vì vậy có khoảng cách là 5.

Hệ thống đường tàu được quản lý bởi một hệ thống máy tính. Không may, sau một sự cố mất điện, máy tính bị mất thông tin các ga tàu nằm ở đâu và khối chứa ga tàu đó thuộc loại nào. Manh mối duy nhất mà hệ thống máy tính còn giữ được là số thứ tự của khối chứa ga tàu 0, và khối này luôn thuộc loại *C*. May mắn là hệ thống máy tính vẫn có thể thực hiện truy vấn tính khoảng cách từ một ga tàu bất kỳ này đến một ga tàu bất kỳ khác. Ví dụ, hệ thống máy tính có thể thực hiện truy vấn 'khoảng cách từ ga tàu 0 đến ga tàu 2 là bao nhiêu?' và nó sẽ trả về giá trị 5.

Nhiệm vụ

Bạn cần cài đặt hàm `findLocation` để xác định số thứ tự khối và loại của khối đối với mỗi ga tàu.

- `findLocation(n, first, location, stype)`
- n : số lượng các ga tàu.
- `first`: Số thứ tự khối chứa ga tàu 0.
- `location`: mảng có độ dài n ; bạn cần gán giá trị số thứ tự của khối chứa ga tàu i cho `location[i]`.
- `stype`: mảng có độ dài n ; bạn cần gán loại khối của ga tàu i cho `stype[i]`: 1 cho loại C và 2 cho loại D.

Bạn có thể gọi hàm `getDistance` để giúp bạn xác định vị trí và loại khối của các ga tàu.

- `getDistance(i, j)` trả về khoảng cách từ ga tàu i đến ga tàu j . `getDistance(i, i)` sẽ trả về 0. `getDistance(i, j)` sẽ trả về -1 nếu i hoặc j nằm ngoài khoảng $0 \leq i, j \leq n - 1$.

Subtasks

Trong tất cả các subtasks, số lượng các khối m không lớn hơn 1,000,000. Trong một vài subtasks số lượt gọi hàm `getDistance` bị hạn chế. Hạn chế này thay đổi theo subtasks. Chương trình của bạn sẽ được đánh giá 'wrong answer' nếu nó sử dụng quá giới hạn này.

subtask	điểm	n	số lần gọi <code>getDistance</code>	ghi chú
1	8	$1 \leq n \leq 100$	không hạn chế	Tất cả ga tàu, trừ ga tàu 0, đều nằm ở khối thuộc loại D.
2	22	$1 \leq n \leq 100$	không hạn chế	Tất cả ga tàu phía đông của ga tàu 0 nằm ở khối thuộc loại D, và tất cả ga tàu phía tây của ga tàu 0 nằm ở khối thuộc loại C.
3	26	$1 \leq n \leq 5,000$	$n(n - 1)/2$	không có thêm hạn chế nào
4	44	$1 \leq n \leq 5,000$	$3(n - 1)$	không có thêm hạn chế nào

Chi tiết cài đặt

Bạn phải nộp duy nhất một file, được đặt tên là `rail.c`, `rail.cpp` hoặc `rail.pas`. File này cài đặt `findLocation` như đã được mô tả ở trên sử dụng các đặc tả sau đây. Bạn cũng cần phải nạp header file `rail.h` khi sử dụng C/C++.

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);
```

Khuôn dạng của `getDistance` được mô tả như sau.

C/C++ program

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

Pascal program

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

Sample grader

Sample grader đọc dữ liệu vào theo khuôn dạng sau:

- dòng 1: số thứ tự subtask
- dòng 2: n
- dòng $3 + i$, ($0 \leq i \leq n - 1$): `styp` $[i]$ (1 cho loại C và 2 cho loại D), `location` $[i]$.

Sample grader sẽ in ra `Correct` nếu `location` $[0] \dots \text{location}[n-1]$ và `styp` $[0] \dots \text{styp}[n-1]$ được tính toán bởi chương trình của bạn khớp với dữ liệu vào khi `findLocation` trả về, và `Incorrect` nếu chúng không khớp.