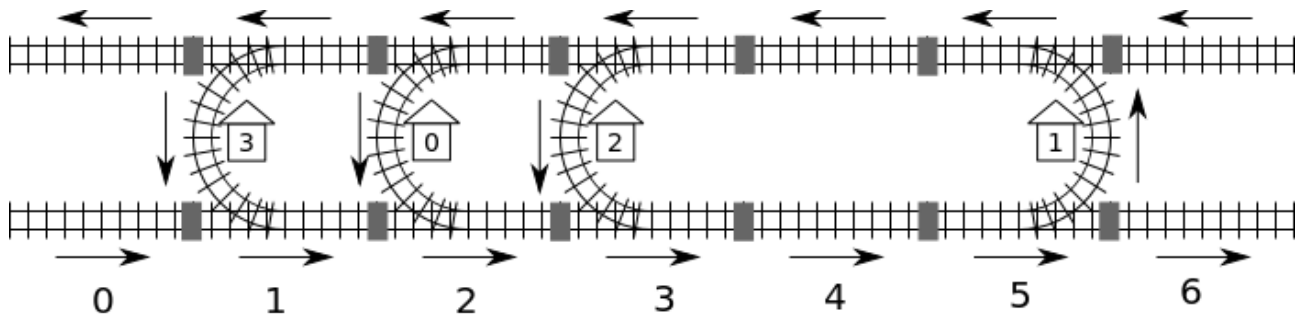


## ریل

تایوان یک خط آهن بزرگ دارد که سواحل غربی و شرقی جزیره را به هم متصل می‌کند. این خط شامل  $m$  بلوک است. این بلوک‌های متوالی به ترتیب از غرب به شرق با شماره‌های  $0$  تا  $m - 1$  شماره گذاری شده‌اند. هر بلوک شامل یک ریل یک طرفه به سمت غرب در شمال و یک ریل یک طرفه به سمت شرق در جنوب است. همچنین در برخی از بلوک‌ها بین ریل‌های شمالی و جنوبی یک ایستگاه قطار واقع شده است.

سه نوع مختلف از بلوک‌ها وجود دارد. بلوک نوع  $C$  یک ایستگاه قطار دارد که باید از ریل شمالی وارد آن شده و از ریل جنوبی از آن خارج شد. بلوک نوع  $D$  یک ایستگاه قطار دارد که باید از ریل جنوبی وارد آن شده و از ریل شمالی از آن خارج شد. در بلوک‌های خالی ایستگاه قطاری وجود ندارد. به عنوان مثال، در شکل زیر بلوک‌های  $0$ ،  $4$  و  $6$  از نوع خالی، بلوک‌های  $1$ ،  $2$  و  $3$  از نوع  $C$ ، و بلوک  $5$  از نوع  $D$  می‌باشد. ریل‌های بلوک‌های مجاور از طریق بست‌هایی که در شکل به صورت مستطیل‌های خاکستری مشخص شده‌اند به یکدیگر وصل می‌شوند.



سیستم ریلی شامل  $n$  ایستگاه است که با شماره‌های  $0$  تا  $n - 1$  شماره گذاری شده‌اند. فرض می‌کنیم که با استفاده از ریل‌ها از هر ایستگاهی به هر ایستگاه دیگری می‌توان رفت. به عنوان مثال، به این طریق می‌توانیم از ایستگاه  $0$  به ایستگاه  $2$  برویم: از بلوک شماره  $2$  شروع کرده، با گذشتن از بلوک‌های  $3$  و  $4$  در ریل جنوبی، از طریق ایستگاه  $1$  به ریل شمالی می‌رویم. سپس با ریل شمالی از بلوک  $4$  عبور می‌کنیم و در نهایت به ایستگاه  $2$  واقع در بلوک  $3$  می‌رسیم.

از آن جایی که مسیرهای مختلفی بین دو ایستگاه وجود دارد، فاصله‌ی یک ایستگاه تا ایستگاه دیگر را برابر با کم‌ترین تعداد بست‌هایی که مسیر از آن‌ها می‌گذرد تعریف می‌کنیم. به عنوان مثال، کوتاه‌ترین مسیر از ایستگاه  $0$  به ایستگاه  $2$  به ترتیب (از چپ به راست) از بلوک‌های  $3 - 4 - 5 - 4 - 3 - 2$  و از تعداد  $5$  بست عبور می‌کند. در نتیجه فاصله‌ی ایستگاه  $0$  تا ایستگاه  $2$  برابر  $5$  است.

یک سامانه‌ی کامپیوتری سیستم ریلی را مدیریت می‌کند. متأسفانه در اثر قطع برق، کامپیوتر دیگر نمی‌داند که ایستگاه‌ها در کدام بلوک‌ها واقع شده‌اند و بلوک مربوط به هر ایستگاه از چه نوعی است. تنها اطلاعاتی که کامپیوتر دارد شماره‌ی بلوک ایستگاه  $0$  است که همیشه از نوع  $C$  می‌باشد. خوش‌بختانه کامپیوتر می‌تواند فاصله‌ی بین هر دو ایستگاه را بپرسد. به عنوان مثال کامپیوتر می‌تواند بپرسد: «فاصله‌ی ایستگاه  $0$  تا ایستگاه  $2$  چقدر است؟» و جواب بگیرد که این مقدار برابر  $5$  است.

## مسئله

شما باید تابع `findLocation` را پیاده سازی کنید که به ازای هر ایستگاه شماره‌ی بلوک و نوع آن را مشخص می‌کند.

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

•  $n$ : تعداد ایستگاه‌ها.

- first: شماره‌ی بلوک ایستگاه ۰.
- location: آرایه‌ای به طول  $n$ ; شما باید شماره‌ی بلوک ایستگاه  $i$  را در `location[i]` قرار دهید.
- stype: آرایه‌ای به طول  $n$ ; شما باید نوع بلوک ایستگاه  $i$  را در `stype[i]` قرار دهید: برای نوع  $C$  از مقدار ۱ و برای نوع  $D$  از مقدار ۲ استفاده کنید.

شما می‌توانید یک تابع `getDistance` را صدا بزنید که به شما کمک می‌کند تا محل و نوع ایستگاه‌ها را بیابید.

- `getDistance(i, j)`: فاصله‌ی ایستگاه  $i$  تا ایستگاه  $j$  را بر می‌گرداند. `getDistance(i, i)` مقدار ۰ را برمی‌گرداند. در صورتی که  $i$  و  $j$  خارج از محدوده‌ی  $0 \leq i, j \leq n - 1$  باشند، `getDistance(i, j)` مقدار  $-1$  را برمی‌گرداند.

## زیرمسئله‌ها

در تمام زیرمسئله‌ها تعداد بلوک‌ها از ۱,۰۰۰,۰۰۰ بیشتر نیست. در بعضی از زیرمسئله‌ها تعداد دفعات فراخوانی تابع `getDistance` محدود است که بر اساس زیرمسئله تغییر می‌کند. در صورتی که برنامه‌ی شما از این محدودیت عبور کند، پیغام 'wrong answer' دریافت می‌کنید.

subtask	points	$n$	getDistance calls	note
1	8	$1 \leq n \leq 100$	unlimited	همه‌ی ایستگاه‌ها به جز ایستگاه ۰ در بلوک‌های نوع $D$ هستند.
2	22	$1 \leq n \leq 100$	unlimited	همه‌ی ایستگاه‌های سمت راست ایستگاه ۰ در بلوک‌های نوع $D$ هستند و همه‌ی بلوک‌های سمت چپ ایستگاه ۰ در بلوک‌های نوع $C$ هستند.
3	26	$1 \leq n \leq 5,000$	$n(n-1)/2$	بدون هیچ محدودیت اضافی
4	44	$1 \leq n \leq 5,000$	$3(n-1)$	بدون هیچ محدودیت اضافی

## جزئیات پیاده‌سازی

شما باید دقیقاً یک فایل با نام `rail.c`، `rail.cpp` یا `rail.pas` را ارسال کنید. این فایل تابع `findLocation` را آن‌گونه که در بالا توضیح داده شد به شکل زیر پیاده‌سازی می‌کند. شما باید هدر فایل `rail.h` را در برنامه‌های  $C/C++$  خود درج کنید.

### برنامه‌ی $C/C++$

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

### برنامه‌ی پاسکال

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

تعریف تابع `getDistance` به شکل زیر است.

### برنامه‌ی C/C++

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

### برنامه‌ی پاسکال

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

### ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

• خط ۱: شماره‌ی زیرمسئله.

• خط ۲:  $n$

• خط  $i + 3$  ( $0 \leq i \leq n - 1$ ): `styp[i]` (۱ برای نوع  $C$  و ۲ برای نوع  $D$ ), `location[i]`.

ارزیاب نمونه در صورتی که `location[0] ... location[n-1]` و `styp[0] ... styp[n-1]` که توسط برنامه‌ی شما محاسبه شده‌اند با ورودی مطابقت داشته باشد، عبارت `Correct` و در غیر این صورت عبارت `Incorrect` را چاپ می‌کند.