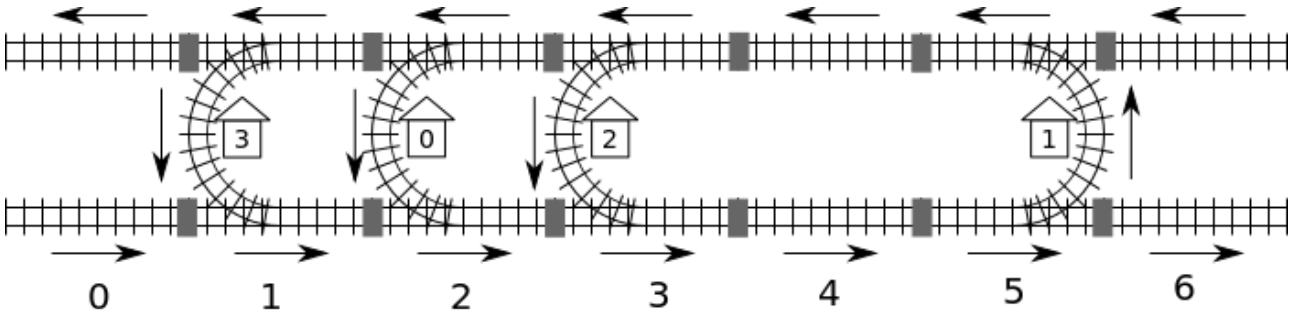


## سكة القطار

لدى تايوان خط سكة حديدي كبير يصل الساحلين الشرقي والغربي للجزيرة. يتألف الخط من مجموعة من  $m$  قطعة. يتم ترقيم القطع المتتالية  $0, \dots, m-1$  بدءاً من النهاية الغربية. كل قطعة لديها مساران واحد شمالي يتجه إلى الغرب وواحد جنوبي يتجه إلى الشرق، كما من الممكن أن تحوي القطعة على محطة قطار بين المسار الشمالي والجنوبي. هناك ثلاثة أنواع مختلفة من القطع، النوع  $C$  يحوي محطة قطار يجب عليك أن تدخلها من المسار الشمالي وأن تخرج منها إلى المسار الجنوبي، النوع  $D$  يحوي على محطة قطار والتي يجب عليك أن تدخلها من المسار الجنوبي وتخرج منها إلى المسار الشمالي. النوع  $empty$  فارغ أي لا يحوي أي محطة قطار. مثلاً في الشكل التالي القطعة رقم  $0$  هي من النوع الفارغ، القطعة رقم  $1$  هي من النوع  $C$ ، والقطعة رقم  $5$  هي من النوع  $D$ . تتصل القطع مع بعضها بشكل أفقي. المسارات للقطع المتجاورة تتصل مع بعضها عن طريق وصلات، تظهر على شكل مستطيلات مظلمة في الشكل التالي.



يحوي نظام القطارات على  $n$  محطة مرقمة من  $0$  وحتى  $n-1$ . لنفترض أنه يمكننا الذهاب من أي محطة إلى أي محطة أخرى باستخدام المسارات الموجودة. مثلاً يمكننا الذهاب من المحطة  $0$  إلى المحطة  $2$  عن طريق الانطلاق من القطعة  $2$ ، ثم المرور بالقطع  $3$  و  $4$  عن طريق المسار الجنوبي، بعد ذلك نمر بالمحطة رقم  $1$  ثم نمر من خلال القطعة رقم  $4$  من المسار الشمالي، وأخيراً نصل إلى المحطة رقم  $2$  في القطعة رقم  $3$ .

بما أنه يمكن أن يوجد أكثر من حل، تعرف المسافة بين محطتين بأنها أقل عدد من الوصلات التي يمر عبرها الطريق، مثلاً إن أقصر طريق من المحطة  $0$  إلى المحطة  $2$  هو عبر  $2-3-4-5-4-3-2$  وهذا يعني العبور عبر  $5$  وصلات لذلك فإن المسافة هي  $5$ .

يتم إدارة نظام القطارات بواسطة نظام حاسوبي، ولسوء الحظ بعد فقدان الطاقة لم يعد بإمكان الحاسوب معرفة أمكنة المحطات وما هي أنواع القطع التي توجد فيها المحطات. الدليل الوحيد الذي يعرفه الحاسوب هو رقم القطعة التي توجد فيها المحطة  $0$ ، والتي ستكون دائماً من النوع  $C$ . لحسن الحظ يمكن للحاسوب حالياً استعمال ومعرفة المسافة بين أي محطة إلى أي محطة أخرى. مثلاً يمكن للحاسوب معرفة "ما هي المسافة بين المحطة  $0$  والمحطة  $2$ " وهي المسافة  $5$ .

## المهمة

يجب عليك تحقيق التابع `findLocation` والذي يحدد من أجل كل محطة رقم القطعة ونوع القطعة.

▪ `(findLocation(n, first, location, stype`

▪ `n`: عدد المحطات.

▪ `first`: رقم القطعة التي تحوي المحطة  $0$ .

▪ `location`: مصفوفة طولها  $n$ ; يجب عليك وضع رقم القطعة التي توجد فيها

المحطة  $i$  في  $location[i]$ .

■  $stype$ : مصفوفة طولها  $n$ ; يجب عليك وضع نوع القطعة التي توجد فيها المحطة  $i$  في  $stype[i]: 1$  من أجل النوع  $C$  و  $2$  من أجل النوع  $D$ .

يمكنك استدعاء التابع  $getDistance$  لمساعدتك على إيجاد أمكنة وأنواع المحطات.

■  $getDistance(i, j)$  يعيد المسافة بين المحطة  $i$  إلى المحطة  $j$ .

$getDistance(i, i)$  سيعيد  $0$ .  $getDistance(i, j)$  سيعيد  $-1$  إذا كان  $i$  أو  $j$  خارج المجال  $0 \leq i, j \leq n - 1$ .

## المهمات الجزئية

في كل المهمات الجزئية سيكون عدد القطع  $m$  ليس أكثر من  $1,000,000$ . في بعض المهمات الجزئية سيكون عدد مرات استدعاء التابع  $getDistance$  محدوداً. يختلف الحد بين المهمات الجزئية. ستحصل على 'wrong answer' إذا تجاوز برنامجك هذا الحد.

المهمة الجزئية	النقاط	$n$	$getDistance$ استدعاءات	ملاحظات
1	8	$1 \leq n \leq 100$	غير محدود	$D$ . كل المحطات ما عدا $0$ تقع في قطع من نوع
2	22	$1 \leq n \leq 100$	محدود	وكل $D$ , كل المحطات على يمين المحطة $0$ هي من النوع $C$ . المحطات على يسار المحطة $0$ هي على قطع من النوع
3	26	$1 \leq n \leq 5,000$	$n(n - 1)/2$	لا يوجد حدود أخرى
4	44	$1 \leq n \leq 5,000$	$3(n - 1)$	لا يوجد حدود أخرى

## تفاصيل التتجيز

يجب عليك إرسال ملف واحد فقط , اسمه `rail.c`, `rail.cpp` أو `rail.pas`.

يحق هذا الملف التابع `findLocation` كما هو مشروع في الأعلى كما يجب عليك تضمين ملف الترويسات `rail.h` من أجل `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);
```

.The signatures of `getDistance` are as follows

### C/C++ program

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

## Pascal program

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

### المصحح التجريبي

يقرأ المصحح التجريبي البيانات وفق الصيغة التالية:

line 1: the subtask number ■

line 2:  $n$  ■

line  $3 + i$ , ( $0 \leq i \leq n - 1$ ): `stype` ■

location[i] (1 for type C and 2 for type D),

سيطبع المصحح التجريبي كلمة 'Correct' إذا كانت الأمكنة صحيحة والأنواع صحيحة كما قمت بإدخالها أو كلمة 'Incorrect' في حال لم تكن صحيحة.