

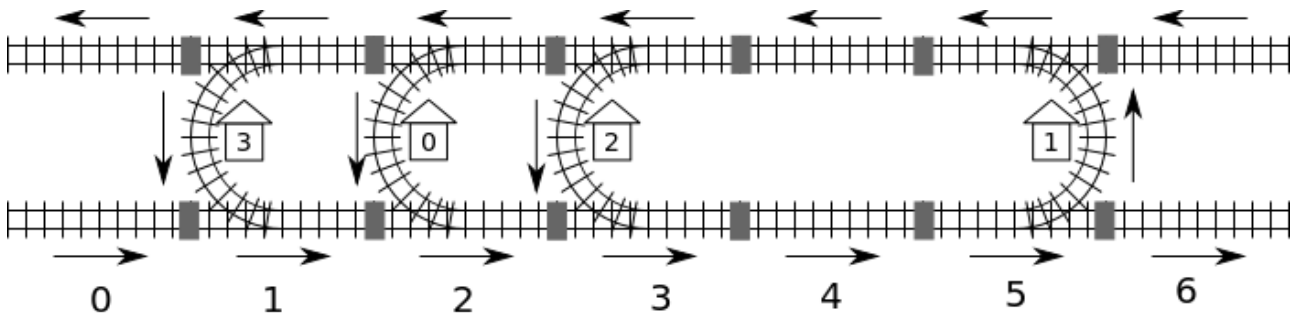


Железная дорога

С запада на восток Тайваня проходит большая железная дорога. Дорога разделена на m зон. Зоны пронумерованы от 0 до $(m - 1)$ с запада на восток. В каждой зоне есть два параллельных пути — северный, по которому движение осуществляется с востока на запад, и южный, по которому поезда движутся с запада на восток. В некоторых зонах расположена железнодорожная станция.

Зоны могут быть трех различных типов. В зоне типа C есть станция, к которой поезд может подъехать по северному пути и выехать по южному. В зоне типа D есть станция, к которой поезд может подъехать по южному пути и выехать по северному. Кроме того, могут быть *пустые* зоны, в которых нет железнодорожной станции.

Например, на рисунке ниже нулевая, четвертая и шестая зоны *пустые*, у первой, второй и третьей зоны — тип C , а у пятой зоны — тип D . Зоны соединены друг с другом по прямой. Соответствующие пути соседних зон связаны друг с другом специальными соединительными устройствами, которые на рисунке изображены серыми прямоугольниками.



Железная дорога объединяет n станций, пронумерованных от 0 до $(n - 1)$. Будем считать, что можно добраться от каждой станции до любой другой, используя железную дорогу. Например, чтобы доехать от нулевой станции до второй, необходимо, начав движение из второй зоны, проехать по южному пути через третью и четвертую зоны, проехать через пятую зону, используя первую станцию, а потом по северному пути через четвертую зону доехать до второй станции, расположенной в третьей зоне.

Так как от одной до другой станции существует несколько маршрутов, то расстояние от одной до другой станции определяется как *минимальное* количество соединительных устройств, которое необходимо проехать. Например, кратчайший маршрут от нулевой до второй станции проходит через зоны 2-3-4-5-4-3 и через пять соединительных устройств, то есть, расстояние равно 5.

Железными дорогами управляет автоматическая система. К сожалению, после внепланового отключения электричества информация о расположении станций и типах зон была потеряна. Однако, известно, что зона, в которой расположена нулевая станция, имеет тип C . К счастью, система умеет отвечать на запрос о расстоянии от станции до станции. Например, на запрос: «Какое расстояние от станции 0 до станции 2 ?», система дает ответ 5.

Постановка задачи

Вам необходимо реализовать функцию `findLocation`, которая определяет для каждой станции номер зоны, в которой она находится, а также тип этой зоны.

- `findLocation(n, first, location, stype)`
 - `n`: количество станций;
 - `first`: номер зоны, в которой расположена станция 0;
 - `location`: массив размера n . Вы должны записать положение i -й станции в `location[i]`;
 - `stype`: массив размера n . Вы должно записать тип зоны, в которой расположена i -ая станция в `stype[i]`. Используйте число 1 для обозначения типа C , число 2 для обозначения типа D .

Для решения задачи вы можете использовать функцию `getDistance`.

- `getDistance(i, j)` возвращает расстояние от станции i до станции j ;
- `getDistance(i, i)` вернет 0;
- `getDistance(i, j)` вернет -1 если i или j не удовлетворяет ограничениям $0 \leq i, j \leq n - 1$.

Подзадачи

Во всех подзадачах количество зон m не превосходит 1 000 000. В некоторых подзадачах ограничено количество вызовов функции `getDistance`. Для разных подзадач ограничение может различаться. Если ваше решение не укладывается в это ограничение, оно получит вердикт 'wrong answer'.

Подзадача	Баллы	n	Вызовы <code>getDistance</code>	Комментарий
1	8	$1 \leq n \leq 100$	не ограничено	Все станции, кроме 0, имеют тип D
2	22	$1 \leq n \leq 100$	не ограничено	Все станции к востоку от станции 0 расположены в зоне типа D , а все станции к западу от станции 0 расположены в зоне типа C .
3	26	$1 \leq n \leq 5\,000$	$n(n - 1)/2$	Дополнительных условий нет.
4	44	$1 \leq n \leq 5\,000$	$3(n - 1)$	Дополнительных условий нет.

Детали реализации

Вы должны послать ровно один файл, названный `rail.c`, `rail.cpp` или `rail.pas`. В этом файле должна быть реализована функция `findLocation` с указанными ниже прототипами. На языках C/C++ вы должны подключить заголовочный файл `rail.h`.

Языки C/C++

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

Язык Pascal

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

У функции `getDistance` будут следующие прототипы.

Язык C/C++

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

Язык Pascal

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

Пример проверяющего модуля

Предоставленный пример проверяющего модуля имеет следующий формат ввода данных:

- строка 1: номер подзадачи
- строка 2: n
- строки $3 + i$, ($0 \leq i \leq n - 1$): `stype[i]` (1 — если зона типа C; 2 — если зона типа D), `location[i]`.

Пример проверяющего модуля выводит сообщение `Correct`, если значения `location[0] ... location[n-1]` и `stype[0] ... stype[n-1]`, посчитанные функцией `findLocation` являются правильными, и `Incorrect` — в противном случае.