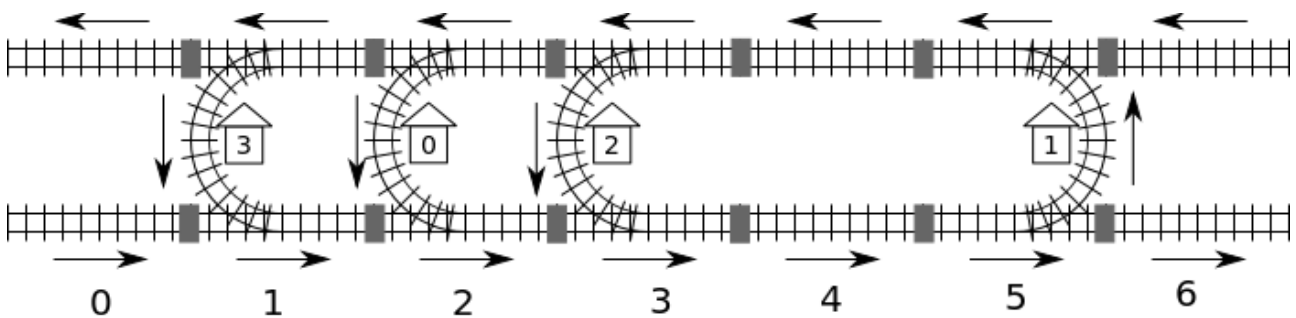


Železnice

Na Tchaj-wanu je dlouhá železniční trasa spojující západní a východní pobřeží ostrova. Trasa se skládá z m úseků. Úseky jsou očíslovány popořadě $0, \dots, m - 1$, od západního konce ostrova. Každý úsek obsahuje na severu jednu jednosměrnou kolej vedoucí směrem na západ a na jihu jednu jednosměrnou kolej vedoucí směrem na východ. Mezi severní a jižní koleji téhož úseku může (ale nemusí) vést spojnice se stanicí.

Existují tři typy úseků. Typ *C* obsahuje stanici, do které musíte přijet ze severní koleje a opustit ji po jižní koleji. Typ *D* obsahuje stanici, do které musíte přijet z jižní koleje a opustit ji po severní koleji. Třetí typ úseku je *prázdný* - v něm žádná stanice není. Například, na následujícím obrázku úseky 0, 4 a 6 jsou prázdné, úseky 1, 2 a 3 jsou typu C, úsek 5 je typu D. Sousední úseky jsou propojeny *spojnicemi*, které jsou na následujícím obrázku vyznačeny šedými obdélníky.



Železniční síť obsahuje n stanic očíslovaných od 0 do $n - 1$. Předpokládáme, že z každé stanice je možné po kolejích dojet do všech ostatních stanic. Například ze stanice 0 můžeme jet do stanice 2 tak, že začneme v úseku 0, projedeme úseky 3 a 4 po jižní koleji, potom projedeme úsekem 5 se stanicí 1, dále pokračujeme úsekem 4 po severní koleji a nakonec se dostaneme do stanice 2 v úseku 3.

Mezi dvěma stanicemi může existovat více různých cest. Vzdálenost z jedné stanice do druhé definujeme jako *minimální* počet spojnic, jimiž musíme projet cestou z jedné do druhé stanice. Například vzdálenost ze stanice 0 do stanice 2 je rovna 5, neboť nejkratší cesta vede přes úseky 2-3-4-5-4-3 a prochází pěti spojnicemi.

O popsanou železniční síť se stará počítačový systém. Naneštěstí po výpadku elektrického proudu, počítač ztratil informace o tom, kde jsou umístěny jednotlivé stanice a jakého typu jsou jednotlivé úseky. Počítači zbyla pouze informace, ve kterém úseku se nachází stanice číslo 0. Úsek se stanicí 0 je vždy typu C. Počítač může klást dotazy na vzdálenost jakýchkoliv dvou stanic. Například se může zeptat „Jaká je vzdálenost ze stanice 0 do stanice 2?“ a dostane odpověď 5.

Úloha

Naprogramujte funkci `findLocation`, která pro každou stanici určí číslo úseku a typ úseku, kde se nachází.

- `findLocation(n, first, location, stype)`
 - `n`: počet stanic.
 - `first`: číslo úseku se stanicí 0.
 - `location`: pole velikosti n ; do něj na pozici i zapište číslo úseku, ve kterém se nachází stanice číslo i .
 - `stype`: pole velikosti n ; do něj na pozici i zapište typ úseku, ve kterém se nachází stanice číslo i .

Můžete volat funkci `getDistance`, která vám pomůže nalézt požadovaný výsledek.

- `getDistance(i, j)` vrací vzdálenost ze stanice i do stanice j . Funkce `getDistance(i, i)` vrátí 0. Volání funkce `getDistance(i, j)` vrátí -1, pokud ji zavoláte s hodnotou parametru i nebo j mimo povolené rozmezí $0 \leq i, j \leq n - 1$.

Podúlohy

Ve všech podúlohách je počet úseků m roven nejvýše 1,000,000. V některých podúlohách je navíc omezen počet volání funkce `getDistance`. Omezení se u jednotlivých podúloh může lišit. Pokud váš program nedodrží uvedené omezení, dostane odpověď 'wrong answer'.

podúloha	počet bodů	n	maximální počet volání <code>getDistance</code>	další omezení
1	8	$1 \leq n \leq 100$	neomezen	Všechny stanice kromě 0 jsou v úsecích typu D.
2	22	$1 \leq n \leq 100$	neomezen	Všechny stanice na východ od stanice číslo 0 jsou v úsecích typu D a všechny stanice na západ od stanice číslo 0 jsou v úsecích typu C.
3	26	$1 \leq n \leq 5,000$	$n(n - 1)/2$	žádná
4	44	$1 \leq n \leq 5,000$	$3(n - 1)$	žádná

Upřesnění implementace

Odevzdejte právě jeden soubor pojmenovaný `rail.c`, `rail.cpp` nebo `rail.pas`. Tento soubor implementuje funkci `findLocation` popsanou výše s následujícími parametry. V případě jazyka C/C++ nezapomeňte vložit hlavičkový soubor `rail.h`.

Program v C/C++

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

Program v Pascalu

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

Parametry funkce `getDistance` jsou následující.

Program v C/C++

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

Program v Pascalu

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

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

- řádek 1: číslo podúlohy
- řádek 2: n
- řádek $3 + i$, ($0 \leq i \leq n - 1$): `styp[i]` (1 pro typ úseku C a 2 pro typ D), `location[i]`.

Ukázkový vyhodnocovač vypíše `Correct`, pokud `location[0] ... location[n-1]` a `styp[0] ... styp[n-1]` vypočtené vaším programem se shodují se vstupem, nebo `Incorrect` v opačném případě.