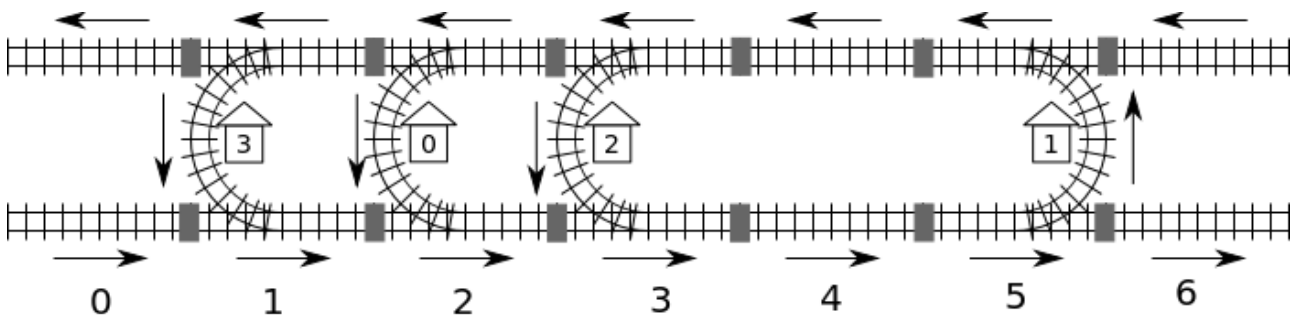




Rail

Tajvan ima veliku železnicu koja povezuje zapadne i istočne delove ostrva. Železnica se sastoji od m blokova. Uzastopni blokovi su numerisani brojevima $0, \dots, m - 1$, počevši od zapadnog kraja. Svaki blok na severu sadrži jednosmernu prugu ka zapadu, dok je na jugu bloka jednosmerna pruga ka istoku. Opciono, unutar bloka između severne i južne pruge može se nalaziti železnička stanica.

Postoje tri vrste blokova. Blok tipa C ima stanicu u koju se ulazi sa severne pruge i iz koje se izlazi na južnu prugu, blok tipa D ima stanicu u koju se ulazi sa južne pruge i iz koje se izlazi na severnu prugu, i blok tipa *prazan*, koji ne sadrži stanicu. Na primer, na sledećoj slici blokovi 0, 4 i 6 su tipa prazan, blokovi 1, 2 i 3 su tipa C , i blok 5 je tipa D . Pruge susednih blokova su spojene *konektorima*, koji su na slici prikazani kao sivi pravougaonici.



Železnički sistem ima n stanica numerisanih od 0 do $n - 1$. Smatraćemo da se *od svake stanice može stići do svake druge stanice* prateći prugu. Na primer, od stanice 0 do stanice 2 možemo stići tako što počnemo od bloka 2, potom prođemo kroz blokove 3 i 4 južnom prugom, potom prođemo blok 5 kroz stanicu 1, potom prođemo kroz blok 4 severnom prugom, i konačno dolazimo u stanicu 2 u bloku 3.

Obzirom na to da postoji više mogućih putanja, rastojanje od jedne stanice do druge definiše se kao *najmanji* broj konektora kroz koje prolazi putanja. Na primer, najkraća putanja od stanice 0 do stanice 2 je kroz blokove 2-3-4-5-4-3 i prolazi kroz 5 konektora, dakle udaljenost je 5.

Železnicom upravlja softverski sistem koji je napisao Dimitrije. Nažalost, zbog greške od jednog karaktera u Dimitrijevom kodu, računar je izgubio podatke o pozicijama stanica, kao i podatke o tome u kom tipu bloka se nalazi svaka stanica. Srećom, Marko je svojim kišobranom zaštitio podatak o broju bloka u kome se nalazi stanica 0, kao i podatak da se ona uvek nalazi u bloku tipa C , tako da su ta dva podatka poznata. Još većom srećom, moguće je vršiti upite o rastojanjima od bilo koje do bilo koje druge stanice. Na primer, moguće je upit 'kolika je udaljenost od stanice 0 do stanice 2?' i na njega će odgovor biti 5.

Zadatak

Treba implementirati funkciju `findLocation` koja za svaku stanicu određuje broj bloka u kome se nalazi, kao i njegov tip.

- `findLocation(n, first, location, stype)`
 - `n`: ukupan broj stanica.
 - `first`: broj bloka koji sadrži stanicu 0.
 - `location`: niz veličine n ; u `location[i]` treba upisati broj bloka u kome se nalazi stanica i .
 - `stype`: niz veličine n ; u `stype[i]` treba upisati tip bloka stanice i : 1 za tip C, i 2 za tip D.

Funkciju `getDistance` možete pozivati da vam pomogne da nađete lokacije stanica i tipove njihovih blokova.

- `getDistance(i, j)` vraća rastojanje od stanice i do stanice j . `getDistance(i, i)` vraća 0. `getDistance(i, j)` će vratiti -1 ako je i ili j van intervala $0 \leq i, j \leq n - 1$.

Podzadaci

U svim podzadacima broj blokova m nije veći od 1,000,000. U nekim podzadacima broj poziva funkcije `getDistance` je ograničen. Ovo ograničenje može biti različito u različitim podzadacima. Vaš program će dobiti odgovor 'wrong answer' ako prekorači ovo ograničenje.

podzadatak	poeni	n	broj <code>getDistance</code> poziva	napomena
1	8	$1 \leq n \leq 100$	neograničen	Sve stanice osim 0 su u blokovima tipa D.
2	22	$1 \leq n \leq 100$	neograničen	Sve stanice istočno od stanice 0 su u blokovima tipa D, sve stanice zapadno od stanice 0 su u blokovima tipa C.
3	26	$1 \leq n \leq 5,000$	$n(n - 1)/2$	nema dodatnih ograničenja
4	44	$1 \leq n \leq 5,000$	$3(n - 1)$	nema dodatnih ograničenja

Detalji implementacije

Treba poslati tačno jedan fajl, nazvan `rail.c`, `rail.cpp` ili `rail.pas`. Ovaj fajl implementira `findLocation` kao što je objašnjeno, koristeći sledeće potpise. U C/C++ implementaciji morate include-ovati header fajl `rail.h`.

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);
```

Potpisi za `getDistance` su sledeći.

C/C++ program

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

Pascal program

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

Opis priloženog grejdera

Priloženi grejder čita ulaz u sledećem formatu:

- linija 1: broj podzadatka
- linija 2: n
- linije $3 + i$, ($0 \leq i \leq n - 1$): `stype[i]` (1 za tip C i 2 za tip D), `location[i]`.

Priloženi grejder će ispisati `Correct` ako se vrednosti `location[0] ... location[n-1]` i `stype[0] ... stype[n-1]` koje je vaš program izračunao slažu sa ulazom grejdera, ili `Incorrect` ako se ne slažu.