

## Problem Brperm

C header: brperm.h  
C++ header: brperm.h

Megjegyzés: a kifejezésekben a  $\overline{b_1 \dots b_k}$  egy egész szám  $k$ -jegyű kettes számrendszerbeli alakját jelenti, ahol  $b_1$  a legnagyobb helyiértékű,  $b_k$  a legkisebb helyiértékű bit.

Roxána úrboszi miközben seprűjén lovagolva körbejárt a galaxisban, rátalált a a BR-PERM bolygóra, ahol egy furcsa táncot járnak. Ebben a táncban a táncosok egy sorban állnak és az adott sorban mindig helyet cserélve, átrendezik magukat a sorban. Ha pontosan  $2^k$  ember táncol, akkor - minden sorszámot  $k$ -jegyű kettes számrendszerbeli számként felírva -, aki a  $\overline{b_1 \dots b_k}$  helyen áll, az áttáncol a  $\overline{b_k \dots b_1}$  helyre (0-tól kezdve a sorszámozást).

Roxána megfigyelte, hogy minden ember a BR-PERM bolygón egyszínű ruhát visel, a lehetséges 26 különböző szín valamelyikét. Ezeket a színeket a Latin ábécé betűivel jelöljük.

A BR-PERM bolygó lakói számára különleges jelentőségűek azok a táncos sorrendek, amiknél a táncosok ruháinak a színe az átrendeződés előtt és után megegyezik egymással. Ezeket a sorokat ők *szép*-nek nevezik. Például,  $k = 2$  esetén a sor négy táncosból áll: 0, 1, 2, 3, akkor a tánc után a sor: 0, 2, 1, 3 sorrendben áll. Így az *abba* színű ruhás emberekből álló sor *szép*, míg a *abca* nem.

A BR-PERM bolygó lakói Roxánát kérik meg, hogy segítsen nekik egy nehéz probléma megoldásában. (Az úrboszorkányoknak mindig segítenie kell az embereknek megoldani a problémájukat.) Megmutatnak neki egy hosszú,  $n$  táncosból álló sort és feltesznek neki néhány kérdést a következő formában: „a  $2^k$  hosszú,  $i$ . táncossal kezdődő sor *szép*?”

## Interakció

Két függvényt kell megvalósítanod. Az első:

```
void init(int n, const char s[]);
```

Ezt a függvényt pontosan egyszer hívjuk meg, az interakció elején. Az  $s$  paraméter megadja a táncosok ruháinak a színét.

A második megvalósítandó függvény:

```
int query(int i, int k);
```

Ezt a függvényt pontosan  $Q$ -szor hívjuk meg és 1-et ad vissza akkor és csak akkor, ha az  $s$  string  $2^k$  hosszúságú folytonos részsorozata az  $i$ . táncostól (0-val kezdve a sorszámozást) kezdve *szép*. Minden más esetben 0-t kell visszaadnia. A tesztesetekben a részsorozat sosem haladja meg a  $s$  string hosszát.

## Korlátok

- $1 \leq N \leq 500\,000$
- $1 \leq Q \leq 500\,000$

## 1. részfeladat (13 pont)

- $1 \leq N \leq 1\,000$
- $1 \leq Q \leq 1\,000$

## 2. részfeladat (37 pont)

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$

## 3. részfeladat (17 pont)

- Az `s` string csak 'a' és 'b' karaktereket tartalmaz.
- A színeket egymástól függetlenül, véletlenszerűen választják, az egyes tesztesetekben előre megadott valószínűséggel.

## 4. részfeladat (33 pont)

- Nincs további korlát.

### Példa

input	output
<code>init(8, "axxyxyb")</code>	<code>query(0, 3) = true</code> <code>query(1, 1) = true</code> <code>query(0, 2) = false</code> <code>query(3, 2) = true</code>