

Feladat Opsir

Bemenet `opsir.in`
Kimenet `opsir.out`

Adott két karakterlánc, az S és a T , amelyek hossza n és m . A karakterláncok kizárólag az angol ábécé kisbetűiből állnak. A betűk sorszámozása 1-től kezdődik.

A T karakterláncon kétféle műveletet lehet végezni:

- 1 p : a p pozíción lévő betű törlődik;
- 2 $st\ dr$ (ahol $st \leq dr$): a $[st, dr]$ intervallumban lévő pozícióknak megfelelő részkarakterlánc betűi növekvő sorrendbe (ábécé sorrendbe) lesznek rendezve;

ahol p , st és dr a T karakterlánc betűinek pozíciói.

Kezdetben, a T sorozat összes betűje szintelen. Egy 2-es típusú művelet csak akkor hajtható végre, ha a $[st, dr]$ intervallumban lévő pozícióknak megfelelő részkarakterlánc összes betűje szintelen. A rendezés után, a részkarakterlánc minden betűje színes lesz.

Követelmények

Minden adott S és T karakterlánc-pár esetén:

1. Írassuk ki azokat a különböző betűket, amelyek legalább az egyik karakterláncban előfordulnak, és mind-egyiknél annak a karakterláncnak a szimbólumát (az S vagy T betűket), amelyben legtöbbször fordul elő. Ha ugyanannyiszor fordul elő, akkor a T szimbólum lesz kiírva.
2. Határozzunk meg 1-es és/vagy 2-es típusú művelet sorokat, amelyeket alkalmazva a T karakterláncra az átalakul S -re. Írassuk ki a DA szöveget, ha létezik ilyen művelet sor, vagy a NU szöveget, ha nem létezik.

Bemeneti adatok

Az `opsir.in` bemeneti állomány első sora egy c természetes számot tartalmaz, a megoldandó követelmény számát (1 vagy 2), minden karakterlánc-párra.

Az állomány második sora a k természetes számot tartalmazza, amely a tesztek számát jelenti. Az állomány következő soraiban minden tesztnek megfelelő három-három sorban a karakterlánc-pár adatai vannak, az alábbiak szerint:

az első sorban az n és az m , ebben a sorrendben, a felhívó szövegnek megfelelően, a második sorban az S karakterlánc, a harmadik sorban pedig a T karakterlánc.

Kimeneti adatok

A $c = 1$ esetén, az `opsir.out` kimeneti állomány tartalmazni fog, minden egyes teszt esetén, egy-egy nr természetes számot, amely azon különböző betűk számát jelöli, amely betűk megjelennek valamelyik karakterláncban, és a következő nr sorban egy-egy ilyen betű, valamint az S vagy a T , annak megfelelően, hogy melyik karakterláncban szerepel többször. A kisbetűk ábécé sorrendben lesznek kiírva. A $c = 2$ esetén, az `opsir.out` kimeneti állománynak k sora lesz, minden sorban egy tesztnek megfelelő válasz (DA vagy NU), a bemeneti állománynak megfelelő sorrendben.

Korlátok

- $1 \leq k \leq 100$
- $1 \leq n \leq m \leq 200\,000$
- Az S típusú karakterláncok hosszának összege nem haladja meg a 200 000-t, k teszt esetén.
- A T típusú karakterláncok hosszának összege, nem haladja meg a 200 000-t, k teszt esetén.

#	Pontszám	Korlátok
1	20	$c = 1$
2	15	$c = 2$ Az S típusú karakterlánc betűi növekvő (ábécé) sorrendben vannak rendezve, az összes teszt esetén.
3	25	$c = 2$ A T típusú karakterláncok átalakíthatóak a megfelelő S típusú karakterláncra kizárólag 1-es típusú művelettel.
4	40	$c = 2$ egyéb korlátozások nélkül.

Példák

opsir.in	opsir.out	Magyarázat
1 3 2 4 cc cbbd 3 2 aab aa 2 2 ac da	3 b T c S d T 2 a T b S 3 a T c S d T	Az első teszt esetén 3 különböző betű van, a követelménynek megfelelően: a b betű többször fordul elő a T -ben, a c többször fordul elő az S -ben, és a d csak a T -ben fordul elő.
2 1 2 2 zx zx	DA	A karakterláncok azonosak anélkül, hogy bármilyen műveletet kellene alkalmazni.
2 2 2 3 ab bca 4 4 bacc cbac	DA NU	<p>Az első teszt esetén a T karakterláncot rendezzük, így "<u>abc</u>" lesz. Ezután töröljük a harmadik betűt, így az S-el egyenlő karakterláncot kapjuk.</p> <p>A második teszt esetén, ha 2-es típusú műveletet alkalmazunk az első 2 betűből álló részkarakterláncra, megkapjuk a "<u>bcac</u>"-t. Tekintettel arra, hogy az első 2 betű színes lett a rendezés után (a színes betűket aláhúzással jelöljük), többé nem alkalmazhatunk 2-es típusú műveletet a "<u>ca</u>" részkarakterláncra. Ezért nem tudjuk átalakítani a T karakterláncot úgy, hogy S-el legyen egyenlő.</p>