

## Problema Opsir

Fișier de intrare      `opsir.in`  
Fișier de ieșire      `opsir.out`

Se consideră o pereche de șiruri de caractere,  $S$  și  $T$ , de lungime  $n$ , respectiv  $m$ , formate exclusiv din litere mici ale alfabetului englez. Pozițiile literelor sunt numerotate în șir începând de la 1.

Sunt două tipuri de operații ce se pot efectua asupra șirului  $T$ :

- 1  $p$  : se șterge litera de pe poziția  $p$ ;
- 2  $st\ dr$  (cu  $st \leq dr$ ) : se sortează crescător (alfabetic) literele din subsecvența ce corespunde intervalului de poziții  $[st, dr]$  ;

unde  $p$ ,  $st$  și  $dr$  sunt poziții ale unor litere din șirul  $T$ .

Inițial, toate literele șirului  $T$  sunt necolorate. O operație de tip 2 poate fi realizată doar dacă toate literele din subsecvența corespunzătoare intervalului de poziții  $[st, dr]$  sunt necolorate. După efectuarea sortării, toate literele din această subsecvență devin colorate.

### Cerințe

Pentru fiecare dintre perechile de șiruri de tipul  $S$  și  $T$  date:

1. Să se afișeze literele distincte care apar în cel puțin unul dintre șiruri și, pentru fiecare dintre acestea, simbolul șirului (literele  $S$  sau  $T$ ) în care apare de mai multe ori. În caz de egalitate, se alege șirul  $T$ .
2. Să se determine o succesiune de operații de tipul 1 și/sau 2 ce pot fi aplicate șirului  $T$ , care să îl transforme într-un șir egal cu  $S$ . Să se afișeze  $DA$  în cazul în care există o astfel de succesiune de operații, sau  $NU$  în caz contrar.

### Date de intrare

Fișierul de intrare `opsir.in` conține pe prima linie numărul natural  $c$ , reprezentând cerința care trebuie rezolvată (1 sau 2) pentru fiecare dintre perechile de șiruri date.

Pe a doua linie a fișierului se găsește numărul natural  $k$ , ce reprezintă numărul de teste. Fiecare test cuprinde datele specifice pentru o pereche de șiruri de tipul celei precizate în enunț, date care se găsesc pe câte trei linii în fișier, astfel: pe prima linie  $n$  și  $m$ , în această ordine, cu semnificația din enunț, pe a doua linie șirul  $S$ , iar pe a treia linie șirul  $T$ .

### Date de ieșire

Pentru  $c = 1$ , fișierul de ieșire `opsir.out` va conține, pentru fiecare test, câte un număr natural  $nr$ , ce reprezintă numărul de litere distincte ce apar în cel puțin unul dintre șiruri, iar pe următoarele  $nr$  linii, câte o astfel de litera, precum și litera mare  $S$  sau  $T$ , corespunzătoare șirului în care apare de mai multe ori. Literele mici vor fi afișate în ordine alfabetică.

Pentru  $c = 2$ , fișierul de ieșire `opsir.out` va conține  $k$  linii, pe fiecare linie aflându-se răspunsul ( $DA$  sau  $NU$ ) corespunzător câte unui test, în ordinea în care acestea se găsesc în fișierul de intrare.

### Restricții

- $1 \leq k \leq 100$
- $1 \leq n \leq m \leq 200\,000$
- Suma lungimilor șirurilor de tip  $S$  din cele  $k$  teste nu depășește 200 000.
- Suma lungimilor șirurilor de tip  $T$  din cele  $k$  teste nu depășește 200 000.

#	Punctaj	Restricții
1	20	$c = 1$
2	15	$c = 2$ șirurile de tip $S$ din fiecare test au literele sortate crescător/alfabetic.
3	25	$c = 2$ șirurile de tip $T$ din fiecare test pot fi transformate în șirul corespunzător de tip $S$ aplicând doar operații de tip 1.
4	40	$c = 2$ fără alte restricții.

## Exemple

opsir.in	opsir.out	Explicații
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	Pentru primul test sunt 3 litere distincte conform cerinței: litera $b$ apare de mai multe ori în $T$ , $c$ apare de mai multe ori în $S$ , iar $d$ apare doar în $T$ .
2 1 2 2 zx zx	DA	Șirurile sunt egale fără a fi necesară aplicarea vreunei operații.
2 2 2 3 ab bca 4 4 bacc cbac	DA NU	<p>Pentru primul test putem sorta întregul șir <math>T</math>, obținând "<u>abc</u>". Putem șterge apoi a treia literă, obținând un șir egal cu <math>S</math>.</p> <p>Pentru al doilea test, dacă aplicăm o operație de tip 2 pentru subsecvența formată din primele 2 litere, obținem "<u>bc</u>ac". Având în vedere că primele 2 litere devin colorate în urma sortării (fapt reprezentat prin subliniere în acest caz), nu mai putem aplica o operație de tip 2 pentru subsecvența "<u>ca</u>". Prin urmare, nu putem transforma șirul <math>T</math> într-un șir egal cu <math>S</math>.</p>