

Hieroglyphs

O echipă de cercetători studiază asemănările între șiruri de hieroglife. Ei reprezintă fiecare hieroglif printr-un întreg nenegativ. Pentru a desfășura studiul ei folosesc următoarele concepte despre șiruri. Pentru un șir dat A , un șir S se numește **subșir** al lui A dacă și numai dacă S poate fi obținut ștergând anumite elemente (eventual niciunul) din A . În tabelul următor sunt câteva exemple de subșiruri ale șirului $A = [3, 2, 1, 2]$.

Subșir	Cum poate fi obținut din A
$[3, 2, 1, 2]$	Nu se șterge niciun element.
$[2, 1, 2]$	$[3, 2, 1, 2]$
$[3, 2, 2]$	$[3, 2, 4, 2]$
$[3, 2]$	$[3, 2, 4, 2]$ or $[3, 2, 4, 2]$
$[3]$	$[3, 2, 4, 2]$
$[]$	$[3, 2, 4, 2]$

Pe de altă parte, $[3, 3]$ sau $[1, 3]$ nu sunt subșiruri ale lui A .

Considerăm două subșiruri de hieroglife, A și B . Un șir S se numește **subșir comun** al lui A și B dacă și numai dacă S este subșir atât pentru A cât și pentru B .

Mai mult, spunem că subșirul U este **subșir comun universal** pentru A și B dacă și numai dacă se îndeplinesc următoarele două condiții:

- U este subșir comun al lui A și B .
- Orice subșir comun al lui A și B este de asemenea subșir al lui U .

Se poate arăta că oricare două șiruri A și B au cel mult un subșir comun universal.

Cercetătorii au găsit două secvențe de hieroglife A și B .

Șirul A are N hieroglife iar șirul B are M hieroglife.

Ajutați cercetătorii să determine un subșir comun universal pentru șirurile A și B sau determinați dacă nu există un astfel de subșir.

Detalii de implementare

Aveți de implementat următoarea funcție.

```
std::vector<int> ucs(std::vector<int> A, std::vector<int> B)
```

- A : un tablou de lungime N care descrie primul șir.
- B : un tablou de lungime M care descrie al doilea șir.
- Dacă există un subșir comun universal pentru A și B , funcția va returna un tablou care reprezintă subșirul comun universal al lor. Altfel, funcția va returna $[-1]$ (un tablou de lungime 1, al cărui singur element este -1).
- Această funcție este apelată o singură dată la fiecare test.

Restricții

- $1 \leq N \leq 100\,000$
- $1 \leq M \leq 100\,000$
- $0 \leq A[i] \leq 200\,000, 0 \leq i < N$
- $0 \leq B[j] \leq 200\,000, 0 \leq j < M$

Subtaskuri

Subtask	Punctaj	Restricții suplimentare
1	3	$N = M$; A și B sunt formate din N întregi <i>distincti</i> cuprinși între 0 și $N - 1$ (inclusiv)
2	15	Pentru orice întreg k , numărul de elemente ale lui A egale cu k plus numărul de elemente ale lui B egale cu k este cel mult 3.
3	10	$A[i] \leq 1$ pentru orice i cu $0 \leq i < N$; $B[j] \leq 1$ pentru orice j cu $0 \leq j < M$
4	16	Există un subșir comun universal pentru A și B .
5	14	$N \leq 3000$; $M \leq 3000$
6	42	Fără restricții suplimentare.

Exemple

Exemplul 1

Considerăm următorul apel.

```
ucs([0, 0, 1, 0, 1, 2], [2, 0, 1, 0, 2])
```

Aici, subșirurile comune pentru A și B sunt următoarele: $[], [0], [1], [2], [0, 0], [0, 1], [0, 2], [1, 0], [1, 2], [0, 0, 2], [0, 1, 0], [0, 1, 2], [1, 0, 2]$ and $[0, 1, 0, 2]$.

Întrucât $[0, 1, 0, 2]$ este un subșir comun al lui A și B , și toate subșirurile comune ale lui A și B sunt subșiruri ale lui $[0, 1, 0, 2]$, funcția trebuie să returneze $[0, 1, 0, 2]$.

Exemplul 2

Considerăm următorul apel.

```
ucs([0, 0, 2], [1, 1])
```

Aici, singurul subșir comun al lui A și B este șirul vid $[]$. Asta înseamnă că funcția trebuie să returneze un tablou fără niciun element $[]$.

Exemplul 3

Considerăm următorul apel.

```
ucs([0, 1, 0], [1, 0, 1])
```

Aici, subșirurile comune pentru A și B sunt $[], [0], [1], [0, 1]$ and $[1, 0]$. Se poate arăta că un subșir universal comun nu există. Așadar, funcția trebuie să returneze $[-1]$.

Grader local

Format de intrare:

```
N M
A[0] A[1] ... A[N-1]
B[0] B[1] ... B[M-1]
```

Format de ieșire:

```
T
R[0] R[1] ... R[T-1]
```

Aici, R este tabloul returnat de `ucs` și T este lungimea sa.