

Problema Caesar Is Back

C header `caesar.h`
C++ header `caesar.h`

Împăratul tău preferat CEZAR s-a întors! El îți va da următoarea problemă. Definește o *transformare 1-step* astfel: o transformare 1-step schimbă un ‘a’ într-un ‘b’, un ‘b’ într-un ‘c’, ..., un ‘y’ într-un ‘z’, și, în final, un ‘z’ într-un ‘a’. Mai mult, pentru orice număr natural k , el definește o *transformare k-step* ca aplicarea transformării 1-step de k ori. De exemplu, o transformare 3-step transformă un ‘a’ într-un ‘d’. Se observă că o transformare 0-step lasă totul neschimbat, cu alte cuvinte schimbă un ‘a’ într-un ‘a’, un ‘b’ într-un ‘b’, și tot așa.

CEZAR îți oferă două șiruri de caractere A și B , fiecare de lungime n . Acestea sunt ambele **indexate de la 0**. Mai mult, el îți oferă q intervale $[l, r]$ cu $0 \leq l \leq r < n$. Pentru fiecare interval $[l, r]$, el vrea să găsească numărul de perechi de forma (x, y) astfel încât $l \leq x \leq y \leq r$ și există un k cu proprietatea că, oricare $x \leq i \leq y$, avem B_i este transformarea k -step a lui A_i .

Spre exemplu, dacă $n = 3$, $A = \text{aac}$, $B = \text{bbc}$, $l = 0$ și $r = 2$ atunci perechile valide sunt $(0, 0)$, $(0, 1)$, $(1, 1)$ și $(2, 2)$. Pentru $(0, 0)$, $(0, 1)$, $(1, 1)$ luăm $k = 1$, iar pentru $(2, 2)$ luăm $k = 0$.

Protocol de interacțiune

Concurentul trebuie să implementeze două funcții:

```
void init(int n, int q, char A[], char B[]);  
long long query(int l, int r);
```

Funcția `init` va fi apelată **exact o dată**, la începutul interacțiunii. Funcția va primi ca parametrii n , q și cele două șiruri de caractere, A și B . După, comisia va apela funcția `query` de q ori. Aceasta va avea ca parametrii l și r , reprezentând o interogare. Concurentul trebuie să returneze un întreg, răspunsul pentru un interval $[l, r]$, descris în enunț.

Atenție! Concurentul trebuie să nu implementeze funcția `main`, și trebuie să includă (`#include`) header-ul `caesar.h`! Concurenții au voie să folosească variabile globale și alte funcții.

Restricții

- $1 \leq n \leq 1\,000\,000$.
- $1 \leq q \leq 1\,000\,000$.
- A și B conțin doar litere mici ale alfabetului englez.

#	Punctaj	Restricții
1	5	$A = \text{aaa} \dots, B = \text{bbb} \dots$
2	9	A și B conțin doar ‘a’ și ‘n’
3	10	$n \leq 100, q \leq 1\,000$
4	15	$n \leq 1\,000, q \leq 300\,000$
5	30	$q \leq 100\,000$
6	31	Fără restricții suplimentare

Exemple

Date de Intrare	Date de Ieșire
init(3, 1, "aac", "bbc") query(0, 2)	4
init(5, 3, "abcde", "bcdyz") query(1, 3) query(0, 2) query(4, 4)	4 6 1
init(20, 20, "aggccdaloxgnakfivqd", "ckjdfgdnszczhpdmilxrh") query(2, 9) query(8, 10) query(2, 11) query(3, 4) query(9, 15) query(6, 12) query(8, 10) query(8, 10) query(2, 5) query(5, 14) query(8, 13) query(5, 11) query(0, 1) query(6, 14) query(0, 5) query(2, 2) query(0, 3) query(9, 14) query(3, 12) query(8, 11)	11 4 14 2 8 8 4 4 4 5 12 7 9 2 10 7 1 4 7 14 5

Explicații

Primul exemplu Pentru intervalul $[0, 2]$ perechile corecte sunt $(0, 0)$, $(0, 1)$, $(1, 1)$ și $(2, 2)$. Pentru primele trei perechi luăm $k = 1$ care transformă litera 'a' în 'b'. Pentru ultima luăm $k = 0$ care lasă litera 'c' neschimbată.

Al doilea exemplu Pentru intervalul $[1, 3]$ avem perechile corecte $(1, 1)$, $(1, 2)$, $(2, 2)$ și $(3, 3)$. Pentru $(1, 1)$, $(1, 2)$ și $(2, 2)$ alegem $k = 1$ care transformă litera 'b' în 'c' și, respectiv, litera 'c' în 'd'. Pentru $(3, 3)$ alegem $k = 21$, deoarece transformă litera 'd' în 'y'. Prin urmare, răspunsul este 4. Pentru intervalul $[0, 2]$ fiecare pereche este validă. Pentru toate alegem $k = 1$, care transformă litera 'a' în 'b', litera 'b' în 'c' și litera 'c' în 'd', respectiv. Prin urmare, răspunsul este 6. Pentru intervalul $[4, 4]$ singura pereche corectă este $(4, 4)$, pentru care alegem $k = 21$, care transformă litera 'e' în 'z'. Prin urmare, răspunsul este 1.