

Zadanie: LEX

Porównywanie leksykograficzne

Dostępna pamięć: 64 MB.

Niech $s = s_1s_2 \dots s_n$ będzie n -literowym słowem, złożonym z małych liter alfabetu angielskiego. Będziemy zajmować się *podstowami* tego słowa, czyli spójnymi fragmentami postaci $s[i..j] = s_is_{i+1} \dots s_j$. Naszym celem jest leksykograficzne porównywanie różnych par takich podsłów.

Powiemy, że słowo u jest *mniej leksykograficznie* (czyli słownikowo) niż słowo v , jeżeli:

- słowo u jest prefiksem właściwym słowa v , tzn. u stanowi początkowy fragment v krótszy niż v , lub
- słowa u i v różnią się na jakiejś pozycji i na pierwszej takiej różniącej je pozycji u zawiera literę mniejszą niż odpowiadająca jej litera w słowie v .

Zapisujemy to jako $u < v$.

Na przykład, słowo $abaab$ jest (leksykograficznie) mniejsze niż $abaababa$, słowo $abaa$ jest mniejsze niż $ababa$, ale ani $abab$ nie jest mniejsze niż $abaab$, ani słowo $abaab$ nie jest mniejsze od $abaab$ (czyli od siebie samego).

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite n oraz m ($1 \leq n, m \leq 300000$), oznaczające długość słowa s oraz liczbę zapytań. Drugi wiersz zawiera n -literowe słowo. Każdy z kolejnych m wierszy zawiera cztery liczby całkowite a, b, c, d ($1 \leq a \leq b \leq n$, $1 \leq c \leq d \leq n$), oznaczające zapytanie o porównanie leksykograficzne słów $s[a..b]$ oraz $s[c..d]$.

Wyjście

Na standardowe wyjście Twój program powinien wypisać m wierszy, z których każdy powinien zawierać jeden znak: '<', '>' lub '=', w zależności od tego, czy pierwsze pod słowo z danego zapytania jest mniejsze czy większe leksykograficznie od drugiego pod słowa, czy też równe temu pod słowu.

Przykład

Dla danych wejściowych:

```
13 3
abaababaabaab
8 13 7 7
6 11 4 6
3 5 11 13
```

poprawnym wynikiem jest:

```
<
>
=
```

Wyjaśnienie do przykładu: W pierwszym zapytaniu rozważamy pod słowa $aabaab$ oraz b , w drugiej — $abaaba$ oraz aba , a w trzeciej — aab oraz aab .