

Język znaczników

Albert napisał program, który stosuje formatowanie tekstu na podstawie klucza. Klucz to ciąg cyfr, a program nakłada formatowanie bazując na rozkładzie bitów w kolejnych cyfrach. Komputer Alberta jest w stanie wyświetlić trzy różne formatowania tekstu: **tekst pogrubiony**, *tekst pochylony* oraz ~~tekst przekreślony~~. Formatowania może się na siebie nakładać. Każda cyfra formatowania jest w zakresie od 1 do 7, czyli ma 3 bity. Możemy teraz powiedzieć, że najmniej znaczący bit (najbardziej z prawej strony) koduje pogrubienie, bit po środku pochylenie, a najbardziej znaczący bit (najbardziej z lewej strony) koduje przekreślenie. Na przykład, dla tekstu **witaj** i klucza 11555 wyświetlone zostanie słowo **witaj**, bo 1 to 001 (czyli pogrubienie), a 5 to 101 (czyli pogrubienie i przekreślenie). Jest jeden problem - formatowanie Alberta jest za trudne do ręcznego pisania. Albert chciałby, aby jego kodowanie działało tak samo, jak znane języki znacznikowe (np. HTML albo Markdown). Więc opracował własne zasady kodowania znacznikowego.

Są 3 znaczniki: `` (zakończenie to ``) robi tekst pogrubiony, `<i>` (zakończenie to `</i>`) robi tekst pochylony, a `<s>` (zakończenie to `</s>`) robi tekst przekreślony. W treści znacznika może być kolejny znacznik, tak aby formatowanie można było nakładać. Albert jest jednak zbyt leniwy, by dostosować program do nowego kodowania, więc prosi Cię, abyś przekonwertował tekst kodowany znacznikami na tekst kodowany liczbami.

Twoim zadaniem jest napisanie rozwiązania, które:

- Wczyta tekst z kodowaniem znacznikowym;
- Przekonwertuje go na tekst kodowany liczbowo;
- Wypisze klucz do zakodowania tekstu liczbowo.

Wejście

W pierwszym wierszu wejścia znajdzie się ciąg wszystkich znaków ascii i znaczników, o długości znacznie nie przekraczającej 10^4 . Możesz założyć, że nigdy znakiem nie będzie `<`, `>` albo `|`, które nie znajdują się w znaczniku.

Wyjście

W pierwszym wierszu wyjścia należy wypisać ciąg cyfr o długości wczytanego słowa, oznaczający klucz zakodowanego formatowania na tekst, gdzie każda cyfra należy do zbioru $\{0, 1, \dots, 7\}$.

Przykład

Wejście:

```
<i>Madre słowa</i> ~ Powiedział <s>Albert Einstein</s> Jan Kowalski
```

Wyjście:

```
222222222220000000000000004444444444444440000000000000
```

Wyjaśnienie do przykładu

Pochylenie to środkowy bit, a więc 2 (010), a przekreślenie to najbardziej znaczący bit, czyli 4 (100).