

# RE-Matching

## Aufgabenstellung

Diese Aufgabe baut auf Aufgabe C auf, wird aber getrennt von ihr bewertet. Ihre Aufgabe ist wie folgt:

1. Überlegen Sie sich auf Papier, Sie die mithilfe der Funktion  $f(r, a)$  aus Aufgabe C, die die Restsprache  $L(r)/a$  bzgl. eines einzelnen Zeichens  $a$  berechnet eine Prozedur  $g(r, u)$  bauen, die die Restsprache  $L(r)/u$  bzgl. eines ganzen Wortes baut.
2. Überlegen Sie sich auf Papier, wie Sie mithilfe dieser Prozedur  $g(r, u)$  entscheiden können, ob ein gegebenes Wort  $w \in \Sigma^*$  von einem regulären Ausdruck erkannt wird, d.h. ob  $w \in L(r)$  ist.
3. Fügen Sie Ihre Implementierung von  $f(r, a)$  aus Aufgabe C in die Klasse `RegexRemainder` des Templates für Aufgabe D ein und vervollständigen Sie die neu hinzugekommenen mit `TODO` markierten Methoden.

## Eingabe

Da es bei einigen von Ihnen auf dem letzten Blatt Probleme mit Unicode-Zeichen gab, wurde die Syntax für reguläre Ausdrücke so angepasst, dass keine Unicode-Zeichen mehr verwendet werden. Die regulären Ausdrücke  $\emptyset$  bzw.  $\epsilon$  werden als `{ }` bzw. `()` eingegeben.

Beliebig viele Testfälle (einer pro Zeile), gefolgt von der Zeile `END`.

Jeder Testfall besteht aus einem regulären Ausdruck  $r$  und mehreren Testwörtern  $w_1$  bis  $w_n$ , jeweils getrennt durch ein Semikolon `;`.

## Ausgabe

Eine Zeile für jeden Testfall. In dieser Zeile gibt es einen Eintrag für jedes Testwort  $w$  von der Form `w:y` bzw. `w:n` je nachdem ob das Wort von  $r$  erkannt wird oder nicht. Die Einträge sind wieder getrennt durch Semikola `;`.

Ihre Ausgabe muss Zeichen für Zeichen der korrekten Lösung entsprechen.

## Beispiele

### Sample Input 1

```
{ } ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
() ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
aa ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
ab ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
ba ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a|b ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a|b ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
b(a|() ) ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a* ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
ba* ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a*b* ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
b*a* ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
(b|() ) *a* ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
(b|() ) *b* ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
a*(b|() ) * ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
b*(b|() ) * ; a;b;ab;ba;aaa;aab;aba;abb;baa;bab;bba;bbb
END
```

### Sample Output 1

:	n;	a:	n;	b:	n;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	y;	a:	n;	b:	n;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	y;	b:	n;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	y;	b:	n;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	n;	b:	n;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	n;	b:	n;	ab:	y;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	n;	b:	n;	ab:	n;	ba:	y;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	y;	b:	y;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	y;	b:	y;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	n;	a:	n;	b:	y;	ab:	n;	ba:	y;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	y;	a:	y;	b:	n;	ab:	n;	ba:	n;	aaa:	y;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	n
:	y;	a:	y;	b:	y;	ab:	n;	ba:	y;	aaa:	y;	aab:	n;	aba:	n;	abb:	n;	baa:	y;	bab:	n;	bba:	n;	bbb:	y
:	y;	a:	y;	b:	y;	ab:	y;	ba:	n;	aaa:	y;	aab:	y;	aba:	n;	abb:	y;	baa:	n;	bab:	n;	bba:	n;	bbb:	y
:	y;	a:	y;	b:	y;	ab:	n;	ba:	y;	aaa:	y;	aab:	n;	aba:	n;	abb:	n;	baa:	y;	bab:	n;	bba:	y;	bbb:	y
:	y;	a:	y;	b:	y;	ab:	n;	ba:	y;	aaa:	y;	aab:	n;	aba:	n;	abb:	n;	baa:	y;	bab:	n;	bba:	y;	bbb:	y
:	y;	a:	n;	b:	y;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	y
:	y;	a:	y;	b:	y;	ab:	y;	ba:	n;	aaa:	y;	aab:	y;	aba:	n;	abb:	y;	baa:	n;	bab:	n;	bba:	n;	bbb:	y
:	y;	a:	n;	b:	y;	ab:	n;	ba:	n;	aaa:	n;	aab:	n;	aba:	n;	abb:	n;	baa:	n;	bab:	n;	bba:	n;	bbb:	y