

Opis

Napisz program w Javie, który będzie realizował następujące operacje:

1. Konwertuje wyrażenia arytmetyczne i instrukcje przypisania z notacji INF do ONP.
2. Konwertuje wyrażenia arytmetyczne i instrukcje przypisania z ONP do notacji INF, używając minimalną liczbę nawiasów, gwarantującą podczas obliczania wyrażenia w INF taką kolejność operacji jak w wyrażeniu ONP.

Instrukcja przypisania ma postać: **operand = wyrażenie arytmetyczne.**

Wyrażenia arytmetyczne mogą zawierać jedynie:

- a. nawiasy: (,) - tylko w notacji INF
- b. operandy: małe litery alfabetu angielskiego
- c. operatory:

Operator	Priorytet	Łączność	Rodzaj operatora
=	0	prawostronna	przypisania
< >	1	lewostronna	relacyjny
+ -	2	lewostronna	addytywny
* / %	3	lewostronna	multiplikatywny
^	4	prawostronna	potęgowania
~	5	prawostronna	unarny

Wejście

Dane do programu wczytywane są ze standardowego wejścia zgodnie z poniższą specyfikacją. Pierwsza linia wejścia zawiera liczbę całkowitą z , oznaczającą liczbę linii zawierających wyrażenia arytmetyczne, których opisy występują kolejno po sobie.

Każda linia zawiera co najmniej 6 znaków i nie przekracza 256 znaków, może mieć jedną z dwóch postaci:

INF: wyrażenie arytmetyczne lub instrukcja przypisania, zapisane w notacji infiksowej

ONP: wyrażenie arytmetyczne lub instrukcja przypisania zapisane w notacji ONP

Przy czym wyrażenia mogą zawierać dowolne znaki. Program najpierw usuwa znaki niewystępujące w wyrażeniach, w tym spacje oraz sprawdza poprawność wyrażień.

Można założyć, że po usunięciu błędnych symboli wyrażenia wejściowe w postaci **INF** są poprawne jeśli są poprawne w C. Natomiast wyrażenia w postaci **ONP** są poprawne jeśli są obliczalne.

Wyjście

- Wyrażenie poprzedzone na wejściu napisem "**INF:** " musi być na wyjściu poprzedzone napisem "**ONP:** " i analogicznie wyrażenie poprzedzone na wejściu napisem "**ONP:** " musi być na wyjściu poprzedzone napisem "**INF:** ". W przypadku błędnego wyrażenia, na wyjściu, zamiast skonwertowanego wyrażenia pojawi napis **error**.
- W przypadku konwersji wyrażenia w ONP do w INF, wyrażenie w INF musi zawierać minimalną liczbę nawiasów, gwarantującą podczas obliczania taką kolejność operacji (uwzględniając typ łączności i priorytety operatorów) jak w wyrażeniu ONP, np. ONP: **xabc**=** zostanie przekształcone do INF: **x=a*(b*c)**

- W przypadku wyrażeń w notacji **INF**, np. **INF**: $(a, + b) / \dots [c3$, program pozostawia jedynie: $(a+b)/c$, pozostałe znaki, w tym spacje – odrzuca, dodatkowo sprawdza poprawność wyrażenia, po czym dokonuje konwersji, wypisując na wyjściu: **ONP**: $ab+c/$.
- W przypadku wyrażeń w notacji **ONP**, np. **ONP**: $(a, b, \dots) . c ; - , *$ program pozostawia jedynie: $abc-*$, dodatkowo sprawdza, czy wyrażenie jest poprawne, po czym dokonuje konwersji, wypisując na wyjściu: **INF**: $a*(b-c)$.

Wymagania implementacyjne

Ogólnie jak w poprzednich programach, w szczególności jedynym możliwym importem jest import skanera wczytywania z klawiatury. Tym samym klasę stosu należy zaimplementować samodzielnie.

Przykład danych

wejście:	wyjście:
10	
ONP: xabc**=	INF: $x=a*(b*c)$
ONP: ab+a~a-+	INF: $a+b+(\sim a-a)$
INF: a+b+(\sim a-a)	ONP: ab+a~a-+
INF: x=~~a+b*c	ONP: xa~~bc*+=
INF: t=~a<x<~b	ONP: ta~x<b~<=
INF: $(a, + b) / \dots [c3$	ONP: ab+c/
ONP: $(a, b, \dots) . c ; - , *$	INF: $a*(b-c)$
ONP: abc++def++g+++	INF: error
INF: x=a=b=c	ONP: xabc===
ONP: xabc===	INF: $x=a=b=c$