

### Opis

Napisz program w Javie, który będzie realizował następujące operacje:

1. Najpierw usuwa znaki, które nie mogą występować w zadanych wyrażeniach, takie jak spacje, przecinki itp. oraz sprawdza poprawność składniową wyrażeń.  
Przy czym można założyć, że po usunięciu zbędnych symboli wyrażenia wejściowe są poprawne w następujących przypadkach:
  - w postaci **INF** - jeśli są poprawne w C.
  - w postaci **ONP** - są jeśli są obliczalne.
2. Konwertuje wyrażenia arytmetyczne i instrukcje przypisania z notacji INF do ONP.
3. Konwertuje wyrażenia arytmetyczne i instrukcje przypisania z ONP do notacji INF, zawierającej minimalną liczbę nawiasów, gwarantującą taką kolejność obliczeń 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 wyrażeń.

### 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 wyrażen w notacji INF, np. INF:  $(a,+b)/..[c3$ , 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 wyrażen w notacji ONP, np. ONP:  $(a,b,.)c;-,*$  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.
- Na końcu kodu przesyłanego submitu proszę dopisać w formie komentarza dane wejściowe, zawierające po 10 przykładowych wyrażen dla każdej formy wyrażenia (INF i ONP) i otrzymane wyniki.

### Przykład danych

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