F: Flamaster

Limit pamięci: **128 MB**

Jasio uwielbia porządek i bardzo irytują go niepospr ątane nawiasy. Ostatnio zobaczył napis `){([`, który go zdenerwował, gdyż nie jest to poprawne nawiasowanie. Szczęśliwie okazało się, że Jasio miał w kieszeni flamaster i mógł dopisać na początku napisu `(`, zaś na końcu `)}` i już mógł spać spokojnie, gdyż ciąg `(){([])}` jest poprawnym nawiasowaniem.

Jasio postanowił, że misją jego życia będzie naprawianie ciągów nawiasów poprzez dopisywanie nawiasów na początku i na końcu zobaczonego ciągu, tak by uzyskany ciąg nawiasów był poprawny. Jako że jego flamaster ma mało atramentu, Jasio chciałby dopisać możliwie mało nawiasów, aby osiągnąć swój cel. Pomóż mu! Napisz program, który: wczyta ciąg nawiasów, obliczy dla niego najkrótszy poprawiony ciąg nawiasów i wypisze go lub stwierdzi, że takiego ciągu nie ma.

Poprawne nawiasowania są zdefiniowane rekurencyjnie w następujący sposób:

- ciąg pusty jest poprawnym nawiasowaniem;
- jeśli S i T to poprawne nawiasowania, to ich złączenie ST też jest poprawnym nawiasowaniem;
- jeśli S to poprawne nawiasowanie, to poprawnymi nawiasowaniami są też `(`S`)`, `[`S`]` oraz `{`S`}`.

Wejście

W pierwszym i jedynym wierszu wejście znajduje się niepusty ciąg składający się ze znaków `(`, `)`, `[`, `]`, `{`, `}` o długości nieprzekraczającej 1 000 000; jest to ciąg nawiasów, który ma naprawić Jasio.

Wyjście

Na wyjściu powinien znaleźć się poprawiony ciąg nawiasów z wejścia. Jeśli istnieje wiele poprawionych nawiasowań o najmniejszej możliwej długości, należy wypisać którekolwiek z nich.

Jeśli naprawienie wejściowego ciągu nawiasów jest niemożliwe, należy wypisać słowo `NIE`.

Przykład

Wejście	Wyjście
`){([`	`(){([])}`

Test odpowiada przykładowi w treści zadania.

Wejście	Wyjście
`([]}`	`NIE`

Podanego ciągu nawiasów nie da się poprawić.