

Lista 3

Zadanie 1 (0,5 punktu). Przetłumacz następujące wyrażenia do postaci ONP. Następnie oblicz je w tej postaci, prezentując w jaki sposób zmienia się stos argumentów.

- $(16 - 8) / (7 + 2)$
- $(22 + 2) * (8 - 2) / 2 * (4 + 2 + 2)$
- $1 - ((1 - 1) - 1 - (1 - 1))$

Zadanie 2 (0,25 punktu). Dla ciągu tokenów S , niech $r(S)$ oznacza ciąg S w odwrotnej kolejności. Załóżmy, że S jest wyrażeniem notacji prefiksowej (polskiej). Czy $r(S)$ musi być wyrażeniem odwrotnej notacji polskiej, reprezentującym to samo obliczenie, co S ?

Zadanie 3 (1 punkt). Wskaż (w sumie) cztery rodzaje sytuacji, w których funkcja `postfix_eval` lub `infix_to_postfix` rzuci wyjątek, przy założeniu, że wejściem funkcji jest (napis reprezentujący) ciąg poprawnych tokenów, ale niekoniecznie reprezentujący poprawne wyrażenie infiksowe lub ONP. Spróbuj zinterpretować, jakim błędem w wyrażeniach odpowiadają te wyjątki. W przypadku `infix_to_postfix` załóż, że została wykonana poprawka z Zadania 4.

Zadanie 4 (0,75 punktu). Napisz funkcję `is_operand(s)`, która dla niepustego napisu s rozstrzyga, czy jest on jednej z następujących postaci:

- Ciąg liter, po którym (bez spacji) następuje ciąg cyfr dziesiętnych, np. "abc123".
- Ciąg cyfr dziesiętnych, np. "0123".
- Ciąg cyfr dziesiętnych poprzedzony minusem, np. "-1001".

Następnie zastąp test `token.isalnum()` w funkcji `infix_to_postfix` przez test `is_operand(token)`.

Zadanie 5 (0,5 punkt). Korzystając z funkcji z wykładu, napisz program – interaktywny kalkulator wyrażeń (w notacji infiksowej). Użytkownik wprowadza wyrażenia w zwykłej notacji, w którym występują tylko liczby całkowite, nawiasy i operatory $+$, $-$, $*$, $/$. Program odpowiada wypisując wartość tego wyrażenia. Program kończy działanie po wpisaniu polecenia 'quit'. Przykładowa interakcja:

```
>>> ( 1 + 2 ) * 3
9
>>> 4 * ( 3 - 1 )
8
>>> quit
```

Nieobowiązkowe: pozwól użytkownikowi na wprowadzanie wyrażeń, w których tokeny nie zawsze są rozdzielone białymi znakami, np "1* (2 +3)".

Zadanie 6 (1 punkt). Zmodyfikuj przykład myjni z wykładu tak, że:

- Myjni jest n , gdzie n to 2 plus ostatnia cyfra Twojego indeksu.
- Nowe samochody wymagają `randrange(1, k)` minut mycia, gdzie k to 5 plus przedostatnia cyfra Twojego indeksu.
- Myjnie myją samochody niezależnie. W każdej minucie każda wolna myjnia powinna przyjąć oczekujący samochód, jeśli taki znajduje się w kolejce.
- Samochód z przodu kolejki w losowy (i uczciwy) sposób wybiera wolną myjnię, do której wjeżdża (o ile jest co najmniej jedna wolna myjnia).

Dla jakiego prawdopodobieństwa pojawienia się samochodu (w każdej minucie) zdolności myjni zostają przekroczone przez przyjeżdżające samochody?