Skúšobné zadanie na druhé cvičenie

2. decembra 2024

1 Infixová notácia (7.5 boda)

Napíšte funkciu `to_infix`, ktorá dostane ako argument číselný výraz v postfixovej notácii a vráti reťazec, ktorý bude v normálnej infixovej notácii. Pritom reťazec, ktorý funkcia vracia má obsahovať každý subvýraz dôsledne ozátvorkovaný, aby sme nemuseli špekulovať o prioritách. Čísla, operácie aj zátvorky majú byť oddelené medzerami.

Uvedomte si, že to je veľmi podobné zadanie ako domáce, aj keď to na to možno na prvý pohľad nevyzerá. Namiesto číselných výsledkov budete na zásobník dávať podvýrazy prevedené do infixovej notácie.

```
>>> to_infix("1 2 *")
'( 1 * 2 )'
>>> to_infix("1 2 * 3 +")
'( ( 1 * 2 ) + 3 )'
>>> to_infix("1 2 3 * +")
'( 1 + ( 2 * 3 ) )'
>>> to_infix("1")
'1'
```

2 `to_postfix` (7.5 boda)

Napíšte funkciu `to_postfix`, ktorá je inverzná k `to_infix`.

```
>>> to_postfix("1")
'1'
>>> to_postfix("( 1 + 2 )")
'1 2 +'
>>> to_postfix("( 10 * ( 1 + 2 ) )")
'10 1 2 + *'
>>> to_postfix("( ( 10 + 1 ) * ( 2 / 3 ) )")
'10 1 + 2 3 / *'
```

2.1 Ako to funguje

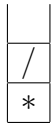
Algoritmus používa zásobník, na ktorom sú iba operácie (t.j reťazce z množiny {"+","−","*","/"}). Po rozdelení argumentu na slová sa pre každé slovo urobí toto (zľava doprava):

1. Ak je slovo rovné "(" , ignoruje sa.
2. Ak je slovo operácia, vloží sa na vrch zásobníka.
3. Ak je slovo rovné ")", vyberie sa operácia z vrchu zásobníka pridá sa k návratovej hodnote.
4. Ak je slovo čokoľvek iné (teda to musí byť číslo), pridá sa k návratovej hodnote.

2.2 Príklad výpočtu

Takto prebieha konverzia pre `"( ( 10 + 1 ) * ( 2 / 3 ) )"`.

slovo	**čo sa deje**	**zásobník**	**návratová hodnota**
(	Ignorujeme	⌊ ⌋	”
(	Ignorujeme	⌊ ⌋	”
10	Pridáme 10 k návratovej hodnote	⌊ ⌋	’10’
+	Pridáme operáciu + do zásobníka	⌊ ⌋ ⌊ + ⌋	’10’
1	Pridáme 1 k návratovej hodnote	⌊ ⌋ ⌊ + ⌋	’10 1’
)	Vyberieme operáciu + zo zásobníka, pridáme k návratovej hodnote	⌊ ⌋	’10 1 +’
*	Pridáme operáciu * do zásobníka	⌊ ⌋ ⌊ * ⌋	’10 1 +’
(	Ignorujeme	⌊ ⌋ ⌊ * ⌋	’10 1 +’
2	Pridáme 2 k návratovej hodnote	⌊ ⌋ ⌊ * ⌋	’10 1 + 2’
/	Pridáme operáciu / do zásobníka	⌊ ⌋ ⌊ / ⌋ ⌊ * ⌋	’10 1 + 2’

slovo	**čo sa deje**	**zásobník**	**návratová hodnota**
3	Pridáme 3 k návratovej hodnote		’10 1 + 2 3’
)	Vyberieme operáciu / zo zásobníka, pridáme k návratovej hodnote		’10 1 + 2 3 /’
)	Vyberieme operáciu * zo zásobníka, pridáme k návratovej hodnote		’10 1 + 2 3 / *’

2.3 Pomôcky

Je rozumné si počas výpočtu vytvárať zoznam reťazcov a nie rovno návratový reťazec, až záver potom použiť `.join(...)`.

3 Záverečný príklad

```
>>> to_infix("1 2 + 3 *")
'( ( 1 + 2 ) * 3 )'
>>> to_postfix(to_infix("1 2 + 3 *"))
'1 2 + 3 *'
>>> to_postfix("( 3 * ( 1 + 2 ) )")
'3 1 2 + *'
>>> to_infix(to_postfix("( 3 * ( 1 + 2 ) )"))
'( 3 * ( 1 + 2 ) )'
>>> eval_expr(to_postfix("( 3 * ( 1 + 2 ) )"))
9
```