

## SKÚŠOBNÉ ZADANIE NA DRUHÉ CVIČENIE

### 1. ODOVZDÁVANIE

- Cez github.
- **Repozitár musí byť privátny (!!!)**
- Pozvite užívateľa `gjenca` do repozitára.

### 2. VYPOČÍTAJTE HODNOTU

Napíšte funkciu `myevald(f,d)`, ktorá dostane ako parameter funkciu `f` v prefixovej reprezentácii a slovník `d`, ktorý zobrazuje všetky premenné použité v `f` na čísla. Ako návratovú hodnotu vráti hodnotu funkcie `f`.

```
>>> myevald(1,{})
1
>>> myevald('x',{ 'x':10})
10
>>> myevald(1,{ 'x':10})
1
>>> myevald(['+', 'x', 1],{ 'x':10})
11
>>> myevald(['+', ['*', 'x', 'y'], 'x'],{ 'x':10, 'y':5})
60
>>> myevald(['/', ['*', 'x', 1.5], 'y'],{ 'x':3.14, 'y':256})
0.0183984375
>>>
```

2.1. **Vrátiť, nie vypisovať.** Hodnotu má funkcia vrátiť, nie vypisovať. Vypisovať nemá nič.

### 3. POMÔCKY ATĎ.

Funkcia `myevald` bude rekurzívna. Základné prípady sú, keď je `f` číslo alebo reťazec. Medziným musí teda vaša funkcia zistiť typ parametra `f` a na základe toho sa rozhodnúť, čo robiť.

### 4. ZJEDNODUŠTE (7 BODOV)

Keďže myderive vracia dosť zbytočne zložité funkcie, pre rozumnú použiteľnosť treba výsledok zjednodušiť. Napíšte funkciu `mysimplify(f)`, ktorá dostane funkciu v prefixovej reprezentácii a vráti tiež funkciu v prefixovej reprezentácii, ale jednoduchšiu.

Implementujte pritom tieto transformácie.

- $x + 0, 0 + x \mapsto x$
- $x.1, 1.x \mapsto x$

- $x.0, 0.x \mapsto 0$
- $x/1 \mapsto x$
- výraz bez premenných (ergo konštanta)  $\mapsto$  jeho hodnota.

Posledný bod v našich podmienkach sa najlepšie vyrieši cez volanie `myeval(f,{})`.

```
>>> mysimplify(["*",1,1])
1
>>> mysimplify(["*",1,"x"])
'x'
>>> mysimplify(["*",["*",1,1],"x"])
'x'
>>> mysimplify(["*",["*",0,1],"x"])
0
>>> myderive(["*",["+",1,"x"],"x"],"x")
['+', ['*', ['+', 0, 1], 'x'], ['*', ['+', 1, 'x'], 1]]
>>> mysimplify(myderive(["*",["+",1,"x"],"x"],"x"))
['+', 'x', ['+', 1, 'x']]
>>> myderive(["*",["+",1,"x"],"x"],"y")
['+', ['*', ['+', 0, 0], 'x'], ['*', ['+', 1, 'x'], 0]]
>>> mysimplify(myderive(["*",["+",1,"x"],"x"],"y"))
0
>>> myderive(["/",["+",1,"x"],"x"],"x")
['/', ['-', ['*', ['+', 0, 1], 'x'], ['*', ['+', 1, 'x'], 1]], ['*', 'x', 'x']]
>>> mysimplify(myderive(["/",["+",1,"x"],"x"],"x"))
['/', ['-', 'x', ['+', 1, 'x']], ['*', 'x', 'x']]
```

## 5. POMÔCKY ATĎ.

- Uvedomte si, že funkcia bude opäť rekurzívna; najprv sa zjednodušia podvýrazy a až na zjednodušené podvýrazy sa aplikujú pravidlá o „lokálnom zjednodušovaní“.
- Poriadne si rozmyslite, čo hovorí predošlý bod skôr ako začnete kódovať.