

1) Vytvořte funkci `filter_numbers` s parametrem, kterým bude list různých datových typů. Funkce list profiluje a vrátí nový list, který bude obsahovat pouze hodnoty typu `int` nebo `float` (funkce `type`). Čísla budou seřazena od nejmenšího po největší (funkce `sort`).

(1b)

Vstup:

```
filter_numbers([1.2, "sdas", 4, [12], 3.4, "12", -3, True, 5, 8.1])
```

Výstup:

```
[-3, 1.2, 3.4, 4, 5, 8.1]
```

2) Vytvořte funkci `check_brackets`, která bere jako parametr řetězec. Funkce vrátí `True`, pokud je výraz v řetězci dobře uzávorkován (tzn. každá závorka je uzavřená), jinak `False`.

(1b)

Vstup:

```
check_brackets("(a+b)/(b*(a+c))")
check_brackets("(a+b))/((b*(a+c)))")
check_brackets("(a+b)/(b*(a+c)")
```

Výstup:

```
True
False
False
```

3) Vytvořte funkci `fun_extrems` s třemi parametry: jiná funkce, u které se předpokládá, že vrací číselnou hodnotu, a 2 číselné hodnoty určující interval. Funkce vrátí dvojici (typu `tuple`), obsahující minimální a maximální hodnotu funkce předané v parametru na zadaném intervalu (interval předpokládejte celočíselný). Můžete použít funkce `min` a `max`.

(1.5b)

Vstup:

```
def f(x):
    return x*(x-2)
fun_extrems(f, -5, 5)
```

Výstup:

```
(-1,35)
```

4) Vytvořte funkci `number_of_letters`, která bere jako parametr řetězec. Funkce vrátí slovník, kde klíčem budou samohlásky vyskytující se v řetězci, hodnotou počet výskytů daného znaku. Řetězec si převedte na malá písmena (funkce `lower`).

(1.5b)

Vstup:

```
number_of_letters('Technical University Ostrava')
```

Výstup:

```
{ 'e': 2, 'i': 3, 'a': 3, 'u': 1, 'y': 1, 'o': 1 }
```