

# Programování 2

---

## 2. cvičení, 24-2-202

---

tags: Programování 2, Čtvrtek 1, Čtvrtek 2

### Farní oznamy

---

1. Tento text a kódy ke cvičení najdete v repozitáři cvičení na <https://github.com/PKvasnick/Programovani-2>.
2. **ReCodEx:** Prosím všechny, aby se zaregistrovali.
  - skupina 10:40 : SIS 15 lidí, ReCodEx 9;
  - skupina 12:20 : SIS 19 lidí, ReCodEx 18.

Můžu vás do ReCodExu přidat také já, ale pak hrozí, že přidám někoho, kdo mezitím změnil skupinu. Opožděnou registrací riskujete znevýhodnění u domácích úkolů - budete mít méně času.

#### Obsah:

- Kvíz
- Načítání a zpracování posloupností (maximum, 2 největší prvek a pod.)

### Na zahřátí

---

Pokud to funguje, nedotýkej se toho.



Dobrá, nebo špatná rada?

## Co dělá tento kód

```
1 jmena = {"Jana", "Pavel", "Pepa", "Franta"}
2 dalsi = {"Eva", "Pepa", "Katka", "Standa"}
3 jmena & dalsi
4 ???
5 jmena and dalsi
6 jmena - dalsi
```

## Seřadte podle data narození

```
1 lide = [
2     ("Jana", "Nováková", 1964),
3     ("Kateřina", "Kocourová", 1962),
4     ("Jozef", "Winkler", 1952),
5     ("Petr", "Suchý", 1968),
6     ("Jan", "Michal", 1951)
7 ]
```

## Maximum a jiné vlastnosti posloupností

Tady si procvičíme úplně jednoduché věci, zčásti také proto, abychom si zopakovali některé postupy, které využijete pro domácí úkoly.

## Načtení posloupnosti z konzole a ze souboru

Typické zadání úlohy v ReCodExu:

Načtěte ze standardního vstupu posloupnost desetinných čísel oddělených znakem nového řádku, ukončenou řádkou s číslem -1 (toto číslo do posloupnosti nepatří)

Generická verze:

```
1 def read_from_console() -> list[float]:
2     a = []
3     while True:
4         line = input()
5         if "-1" in line:
6             break
7         a.append(float(line))
8     return a
```

Poznámky:

1. Kód `-> list[float]` oznamuje Pythonu, že výstupem funkce je seznam desetinných čísel. Podobně můžeme oznámit i typy parametrů, a interpret pak zahlásí chybu, pokud někde použijeme nesprávný typ.
2. Logika: Máme nekonečný cyklus a vyskakujeme z něj, pokud narazíme na znak konce posloupnosti. Už jsme si ukazovali, že to jde udělat i šikovněji.
3. Testujeme `"-1" in line` namísto `line == "-1"` Proč? (Už vás někdy ReCodEx trýznil hláškami "End-of-file error" u testů?)
4. Načítáme do paměti celou posloupnost - to nemusí být praktické.

Ze souboru:

```
1 def read_from_file(filename: str) -> list[float]:
2     a = []
3     with open(filename, "r") as infile:
4         for line in infile:
5             if "-1" in line:
6                 break
7             a.append(float(line))
8     return a
```

## Načítání pomocí generátoru

Pokud chceme hledat například maximum posloupnosti a nechceme ji celou načítat, musíme kód pro načítání a hledání promíchat. To je nešťastné, pokud chceme pro zpracování posloupnosti použít stejný kód pro načítání ze standardního vstupu nebo souboru.

```
1 m = float("-inf")
2 while "-1" not in (line := input()):
3     number = float(line)
4     if number > m:
5         m = number
6
```

Poznámky:

- Inicializace: `float("-inf")` je nejmenší číslo v plovoucí desetinné čárce. Jaké je nejmenší celé číslo?

```
1 def read_from_console():
2     while "-1" not in (line := input()):
3         yield float(line)
4     return
5
6 m = float("-inf")
7 for number in read_from_console():
8     if number > m:
9         m = number
10 print(m)
```

Napište kód, který takto nalezne maximum při načítání posloupnosti ze souboru.

## Reduce

Uměli bychom uzavřít otevřený cyklus `while`, resp. `for`, který máme v tomto kódu?

Můžeme použít funkci `functools.reduce`, která dělá přibližně toto:

```
1 def reduce(function, iterable, initializer=None):
2     it = iter(iterable)
3     if initializer is None:
4         value = next(it)
5     else:
6         value = initializer
7     for element in it:
8         value = function(value, element)
9     return value
```

Tedy funkce `reduce` propaguje a aktualizuje nějaký stav přes posloupnost.

```
1 from functools import reduce
2
3 def read_from_console():
4     while "-1" not in (line := input()):
5         yield float(line)
6     return
7
8 maximum = reduce(max, read_from_console, float("-inf"))
9 print(maximum)
```

Takovýto kód bude rychlý, protože cyklus se vykonává uvnitř funkce, a tedy běží v C a ne v Pythonu.

## Podobné úlohy

- rozhodnout, zda je posloupnost čísel monotonní a jak (konstantní, rostoucí, neklesající, klesající, nerostoucí)
- v posloupnosti čísel nalézt druhou největší hodnotu a počet jejích výskytů
- v posloupnosti čísel určit délku nejdelšího souvislého rostoucího úseku

- v posloupnosti čísel určit počet různých hodnot
- v posloupnosti čísel nalézt souvislý úsek se součtem  $K$  (pro zadanou hodnotu  $K$ )
- v posloupnosti kladných čísel nalézt souvislý úsek se součtem  $K$  (pro zadanou hodnotu  $K$ )
- v posloupnosti čísel nalézt souvislý úsek s maximálním součtem.

## 2D úlohy

Máme matici  $r \times s$ , třeba s celými čísly. Chceme

- v matici nalézt maximální podmatici tvořenou kladnými čísly (podmaticí rozumíme souvislý obdélníkový výřez, maximální podmatice znamená podmatice tvořená co nejvíce prvky)
- v matici nalézt podmatici s maximálním součtem prvků.