

Cvícenie 2

Instrukcie:

- Vypracujte všetky úlohy. Na cvícení sa pokúste vypracovať čo najviac úloh a úlohy, ktoré nestihnúte na cvícení, potom vypracujte doma.
- **POZOR!** Všímajte si, či funkcia, ktorú máte definovať, má niečo vracať alebo či má niečo vypisovať! V prípade, že má funkcia niečo **vrátiť**, musí v nej byť použitý príkaz **return**!
- **V prípade, že sa na niektorej úlohe zaseknú, pýtajte sa cvičiaceho alebo odborného praktika.**

Cast 1: funkcie

1A. Vytvorte skript, ktorý definuje funkciu *tretia_mocnina*, ktorá pre vstupný argument *a* **vráti** hodnotu $a*a*a$.

1B. Upravte skript z úlohy 1A tak, aby skript pomocou funkcie *tretia_mocnina* vypísal tretiu mocninu čísla 5.

2A. Vytvorte skript, ktorý definuje funkciu *priemer_troch*, ktorá pre vstupné argumenty *a, b, c* **vráti** hodnotu $(a+b+c)/3$.

2B. Upravte skript z úlohy 2A tak, aby skript pomocou funkcie *priemer_troch* vypísal priemer čísel 5, 6, 7.

Cast 2: funkcie a for cykly

V úlohách 1 až 4 sa budem odvolávať na nasledujúci skript:

```
1 def vypis(n) :  
2     for i in range(n) :  
3         print(i)
```

Funkcia *vypis(n)* definovaná v tomto skripte vypíše čísla od 0 po $n-1$.

1. Vytvorte skript, ktorý definuje funkciu s parametrom *n*, ktorá **vypíše** čísla od 1 po *n*. Váša funkcia by mala vyzeráť ako funkcia *vypis(n)* na obrázku vyššie a od funkcie *vypis(n)* by sa mala líšiť iba v riadku číslo 3.

2. Vytvorte skript, ktorý definuje funkciu s parametrom *n*, ktorá **vypíše** prvých *n* čísel vadsích ako 100. (Pre $n=3$ funkcia vypíše čísla 101, 102 a 103.) Váša funkcia by mala vyzeráť ako funkcia *vypis(n)* na obrázku vyššie a od funkcie *vypis(n)* by sa mala líšiť iba v riadku číslo 3.

3. Vytvorte skript, ktorý definuje funkciu s parametrom *n*, ktorá **vypíše** prvých *n* párnych čísel (začínajúc dvojkou). Urobte to bez použitia podmienok (if-ov). Váša funkcia by mala vyzeráť ako funkcia *vypis(n)* na obrázku vyššie a od funkcie *vypis(n)* by sa mala líšiť iba v riadku číslo 3.

4. Vytvorte skript, ktorý definuje funkciu s parametrom *n*, ktorá **vypíše** klesajúcu postupnosť čísel od *n* po 1. Váša funkcia by mala vyzeráť ako funkcia *vypis(n)* na obrázku vyššie a od funkcie *vypis(n)* by sa mala líšiť iba v riadku číslo 3.

5. Vytvorte skript, ktorý definuje funkciu s parametrom *n*, ktorá **vráti** súčet $1^2 + 2^2 + 3^2 + \dots + n^2$. (Pre tento súčet síce existuje vzorec, ale od Vás teraz chcem, aby ste to zráťali pomocou for-cyklu.)

6. Definujte funkciu, ktorá **vypíše** prvých N členov **aritmetickej postupnosti** $a_{i+1} = a_i + d$ na základe parametrov a_0 , d a N . Čiže na základe parametrov a_0 , d a N funkcia vypíše postupnosť čísel $a_0, a_1, a_2, \dots, a_{N-1}$, v ktorej platí, že $a_{i+1} = a_i + d$. Napríklad pre hodnoty parametrov $a_0=0, d=3, N=5$, funkcia vypíše 0,3,6,9,12.

7. Teraz vyriešte úlohu 6 tak, že v definícii funkcie nepoužijete operáciu $*$ (t.j. operáciu násobenia).

8. Pozrite si skript *meranie_casu.py*, ktorý nájdete na webstránke predmetu. V tomto skripte definujem funkciu *arit*, ktorá rieši úlohu 6 a využíva operáciu $*$. V skripte takisto definujem funkciu *arit2*, ktorá tiež rieši úlohu 6, ale operáciu $*$ už nepoužíva. V skripte potom porovnávam rýchlosť týchto funkcií. Spustíte skript *meranie_casu.py*. Budete si môcť všimnúť, že funkcia *arit2* je rýchlejšia ako funkcia *arit*.

9. Definujte funkciu, ktorá **vypíše** prvých N členov **geometrickej postupnosti** $a_{i+1} = a_i * r$ na základe parametrov a_0 , r a N . Čiže na základe parametrov a_0 , r a N funkcia vypíše postupnosť čísel $a_0, a_1, a_2, \dots, a_{N-1}$, v ktorej platí, že $a_{i+1} = a_i * r$. Napríklad pre hodnoty parametrov $a_0=1, r=2, N=5$, funkcia vypíše 1,2,4,8,16.

Pomocka: umocňovanie sa v pythone robí pomocou symbolov $**$. Napríklad $2**3$ je 8.

10. Teraz vyriešte úlohu 9 tak, že v definícii funkcie nepoužijete operáciu umocnenia.

11. Ak sa funkcie, ktoré ste vytvorili v úlohách 9 a 10, od seba líšia, porovnajte ich rýchlosť. Na porovnanie rýchlostí funkcií sa inšpirujte skriptom *meranie_casu.py*, ktorý nájdete na webstránke predmetu. Pri porovnávaní rýchlostí môžete funkcie spustiť napríklad s parametrami $a_0=1, r=2, N=50\,000$. Pri porovnávaní rýchlostí zakomentujte volania funkcie *print*! Ak by vám niektorý výpočet trval príliš dlho, stlačte **Ctrl+C**. Výpočet tým prerušíte.

12. Definujte funkciu, ktorá **vypíše** prvých N členov **geometrickeho radu** na základe parametrov a_0 , r a N . (**Geometrický rad** je definovaný tak, že i -ty člen radu predstavuje súčet prvých i členov geometrickej postupnosti.) Napríklad pre hodnoty parametrov $a_0=1, r=2, N=5$, funkcia vypíše 1, 3 ($=1+2$), 7 ($=1+2+4$), 15 ($=1+2+4+8$), 31 ($=1+2+4+8+16$). Definujte funkciu tak, aby jej definícia obsahovala iba jeden *for* cyklus.

13. Ak r je v absolútnej hodnote menšie ako 1, tak s rastúcim N sa členy geometrickeho radu blížia k hodnote

$$a_0/(1-r).$$

Tomuto hovoríme, že pre r v absolútnej hodnote menšej ako 1 geometrický rad konverguje k hodnote $a_0/(1-r)$. Napríklad geometrický rad s parametrami $a_0=1, r=0.5$ konverguje k hodnote 2. Otestujte, či sa čísla, ktoré vypíše vaša funkcia z úlohy 12 pre parametre $a_0=1, r=0.5, N=100$, blížia k hodnote 2. Ak nie, potom máte niekde chybu.

14. Teraz vyriešte úlohu 12 tak, že v definícii funkcie nepoužijete operáciu umocnenia. Definícia by stále mala obsahovať iba jeden *for* cyklus.

15. Porovnajte rýchlosť vašich funkcií z úloh 12 a 14 a rýchlosť funkcie, ktorú nájdete v skripte *neefektivny_skript.py* na webstránke predmetu. Funkcia v tomto skripte tiež rieši úlohu 12, ale robí to pomocou dvoch do seba vnorených *for* cyklov. Na porovnanie rýchlostí funkcií sa opäť inšpirujte skriptom *meranie_casu.py*, ktorý nájdete na webstránke predmetu. Ak budete porovnávať s funkciou zo skriptu *neefektivny_skript.py*, môžete napríklad použiť parametre $a_0=1, r=2, N=1000$. Ak budete porovnávať iba funkcie z úloh 12 a 14, potom môžete napríklad použiť parametre $a_0=1, r=2, N=50000$. Pri porovnávaní rýchlostí zakomentujte volania funkcie *print*! Ak by vám niektorý

vypocet trval priliz dlho, stlacte Ctrl+C . Vypocet tym prerusite.

16. Vytvorte skript, ktory pomocou for-cyklov a funkcie print definuje funkciu *grid* s parametrom *n*, ktora do konzoly vykresli mriezku s *n* riadkami a *n* stlpcami. *Vsimnite si pomocku uvedenú nižšie!* Mriezka s dvomi riadkami a dvomi stlpcami by mala vyzerat nasledovne:

```
+ - - - - + - - - - +
|           |           |
|           |           |
|           |           |
|           |           |
+ - - - - + - - - - +
|           |           |
|           |           |
|           |           |
|           |           |
+ - - - - + - - - - +
```

Pomocka: funkcia print automaticky vypisuje na novy riadok. Toto nastavenie sa ale da zmenit zmenou hodnoty parametra end. Napríklad prikazy:

```
print('+', end=' ')
print('-')
```

vypisu:

+ -

Dalsi priklad:

Prikazy:

```
print('+', end=',')
print('-')
```

Vypisu:

+, -