Cvicenie 2

Instrukcie:

- Vypracujte vsetky ulohy. Na cviceni sa pokuste vypracovat co najviac uloh a ulohy, ktore nestihnete na cviceni, potom vypracujte doma.
- Ak sa vam na cviceni podari vyriesit ulohu 15, ohlaste sa cviciacemu, odprezentujte mu vase riesenie a ak to budete mat dobre, dostanete jeden **bonusovy bod**.
- **POZOR!** Vsimajte si, ci funkcia, ktoru mate definovat, ma nieco vracat alebo ci ma nieco vypisovat! V pripade, ze ma funkcia nieco **vratit,** musi v nej byt pouzity prikaz ***return*!**
- **V pripade, ze sa na niektorej ulohe zaseknete, pytajte sa cviciaceho.**

Cast 1: funkcie

1A.Vytvorte skript, ktory definuje funkciu *tretia_mocnina*, ktora pre vstupny argument *a* **vrati** hodnotu *a*a*a* .
1B. Upravte skript z ulohy 1A tak, aby skript pomocou funkcie *tretia_mocnina* vypisal tretiu mocninu cisla 5.

2A.Vytvorte skript, ktory definuje funkciu *priemer_troch*, ktora pre vstupne argumenty *a,b,c* **vrati** hodnotu *(a+b+c)/3*.
2B. Upravte skript z ulohy 2A tak, aby skript pomocou funkcie *priemer_troch* vypisal priemer cisel 5,6,7.

Cast 2: funkcie a for cykly

V ulohach 1 az 4 sa budem odvolavat na nasledujuci skript:

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

Funkcia vypis(n) definovana v tomto skripte vypise cisla od 0 po n-1.

1. Vytvorte skript, ktory definuje funkciu s parametrom n, ktora **vypise** cisla od 1 po n. Vasa funkcia by mala vyzerat ako funkcia vypis(n) na obrazku vyssie a od funkcie vypis(n) by sa mala lisit iba v riadku cislo 3.

2. Vytvorte skript, ktory definuje funkciu s parametrom n, ktora **vypise** prvych n cisel vacsich ako 100. (Pre n=3 funkcia vypise cisla 101,102 a 103.) Vasa funkcia by mala vyzerat ako funkcia vypis(n) na obrazku vyssie a od funkcie vypis(n) by sa mala lisit iba v riadku cislo 3.

3. Vytvorte skript, ktory definuje funkciu s parametrom n, ktora **vypise** prvych n parnych cisel (zacinajuc dvojkou). Urobte to bez pouzitia podmienok (if-ov). Vasa funkcia by mala vyzerat ako funkcia vypis(n) na obrazku vyssie a od funkcie vypis(n) by sa mala lisit iba v riadku cislo 3.

4. Vytvorte skript, ktory definuje funkciu s parametrom n, ktora **vypise** klesajucu postupnost cisel od n po 1. Vasa funkcia by mala vyzerat ako funkcia vypis(n) na obrazku vyssie a od funkcie vypis(n) by sa mala lisit iba v riadku cislo 3.

5. Vytvorte skript, ktory definuje funkciu s parametrom n, ktora **vrati** sucet $1^2+2^2+3^2+\ldots+n^2$. (Pre tento sucet sice existuje vzorec, ale od Vas teraz chcem, aby ste to zratali pomocou for-cyklus.)

6. Definujte funkciu, ktora **vypise** prvych N členov **aritmetickej postupnosti** a_{i+1} = a_{i} + d na základe parametrov a_{0}, d a N. Cize na zaklade parametrov a_{0}, d a N funkcia vypise postupnost cisel a_{0}, a_{1}, a_{2}, ... , a_{N-1}, v ktorej plati, ze a_{i+1} = a_{i} + d. Napriklad pre hodnoty parametrov a_{0}=0, d=3, N=5, funkcia vypise 0,3,6,9,12.

7.Teraz vyrieste ulohu 6 tak, ze v definicii funkcie nepouzijete operaciu * (t.j. operaciu nasobenia).

8. Definujte funkciu, ktora **vypise** prvych N členov **geometrickej postupnosti** a_{i+1} = a_{i} *r na základe parametrov a_{0}, r a N. Cize na zaklade parametrov a_{0}, r a N funkcia vypise postupnost cisel a_{0}, a_{1}, a_{2}, ... , a_{N-1}, v ktorej plati, ze a_{i+1} = a_{i} *r. Napriklad pre hodnoty parametrov a_{0}=1, r=2, N=5, funkcia vypise 1,2,4,8,16.
Pomocka: umocnovanie sa v pythone robi pomocou symbolov ** . Napriklad 2**3 je 8.

9.Teraz vyrieste ulohu 8 tak, ze v definicii funkcie nepouzijete operaciu umocnenia.

10. Ak sa funcie, ktore ste vytvorili v ulohach 8 a 9, od seba lisia, porovnajte ich rychlost. Na porovnanie rychlosti funkcii sa inspirujte skriptom *meranie_casu.py*, ktory najdete na webstranke predmetu. Pri porovnavani rychlosti mozete funkcie spustit napriklad s parametrami a_{0}=1, r=2, N=50 000. Pri porovnavani rychlosti zakomentujte volania funkcie print! Ak by vam niektory vypocet trval priliz dlho, stlacte Ctrl+C . Vypocet tym prerusite.

11. Definujte funkciu, ktora **vypise** prvych N členov **geometrickeho radu** na základe parametrov a_{0}, r a N. (**Geometricky rad** je definovany tak, ze i-ty člen radu predstavuje súčet prvých i členov geometrickej postupnosti.) Napriklad pre hodnoty parametrov a_{0}=1, r=2, N=5, funkcia vypise 1, 3 (=1+2), 7 (=1+2+4), 15 (=1+2+4+8), 31 (=1+2+4+8+16). Definujte funkciu tak, aby jej definicia obsahovala iba jeden for cyklus.

12. Ak r je v absolutnej hodnote mensie ako 1, tak s rastucim N sa cleny geometrickeho radu blizia k hodnote

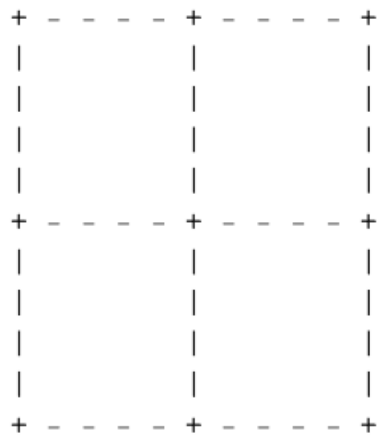
a_{0}/(1-r).

Tomuto hovorime, ze pre r v absolutnej hodnote mensej ako 1 geometricky rad konverguje k hodnote a_{0}/(1-r). Napriklad geometricky rad s parametrami a_{0}=1, r=0.5 konverguje k hodnote 2. Otestujte, ci sa cisla, ktore vypise vasa funkcia pre parametre a_{0}=1, r=0.5, N=100, blizia k hodnote 2. Ak nie, potom mate niekde chybu.

13.Teraz vyrieste ulohu 11 tak, ze v definicii funkcie nepouzijete operaciu umocnenia. Definicia by stale mala obsahovat iba jeden for cyklus.

14. Porovnajte rychlost vasich funkcii z uloh 11 a 13 a rychlost funkcie, ktoru najdete v skripte *neefektivny_skript.py* na webstranke predmetu. Funkcia v tomto skripte tiez riesi ulohu 11, ale robi to pomocou dvoch do seba vnorenych for cyklov. Na porovnanie rychlosti funkcii sa opat inspirujte skriptom *meranie_casu.py*, ktory najdete na webstranke predmetu. Ak budete porovnavat s funkciou zo skriptu *neefektivny_skript.py* , mozete napriklad pouzit parametre a_{0}=1, r=2, N=1000. Ak budete porovnavat iba funkcie z uloh 11 a 13, potom mozete napriklad pouzit parametre a_{0}=1, r=2, N=50000. Pri porovnavani rychlosti zakomentujte volania funkcie print! Ak by vam niektory vypocet trval priliz dlho, stlacte Ctrl+C . Vypocet tym prerusite.

15. ***Uloha za 1 bonusovy bod.*** 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 uvedenu nizsie!* 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. Napriklad prikazy:

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

vypisu:
+ -

Dalsi priklad:
Prikazy:

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

Vypisu:
+,-