

Cvícenie 4

Instrukcie:

- Vypracujte všetky ulohy. Na cvícení sa pokúste vypracovať čo najviac uloh a ulohy, ktoré nestihnete na cvícení, potom vypracujte doma.
- **Špeciálne dávam do pozornosti ulohu číslo 10, ktorá je tážšia ako ostatné a mali by ste k nej prísť ako k výzve!**
- **V prípade, že sa na niektorej ulohy zaseknete, pýtajte sa cvičiaceho.**

Upozornenie: Cvícenie má až 3 strany a pozostáva z 19 uloh!

Sekcia 1:

1. Vytvorte interaktívnu verziu funkcie *vypis(n)* z Cvičenia 2. (podobne ako na slajdoch z prednášky)

2. Definujte funkciu *test_parity* s parametrom *n*, ktorá **vráti** hodnotu `True`, ak je *n* párne číslo, a ktorá **vráti** hodnotu `False`, ak je *n* nepárne. (použite operátor `%`)

Napríklad: volanie *test_parity(0)* **vráti** hodnotu `True`; volanie *test_parity(1)* **vráti** `False`; volanie *test_parity(2)* **vráti** `True`.

3. Definujte funkciu *minimum_dvoch*, ktorá bude mať dva parametre (čísla *a,b*), a ktorá **vráti** minimum z týchto dvoch čísel.

Napríklad: volanie *minimum_dvoch(2,5)* **vráti** hodnotu 2; volanie *minimum_dvoch(2,-3)* **vráti** -3; volanie *minimum_dvoch(10,0)* **vráti** 0.

4. Definujte funkciu *minimum_troch*, ktorá bude mať tri parametre (čísla *a,b,c*), a ktorá **vráti** minimum z týchto troch čísel.

Napríklad: volanie *minimum_troch(2,5,8)* **vráti** hodnotu 2; volanie *minimum_troch(2,-3,0)* **vráti** -3; volanie *minimum_troch(10,3,1)* **vráti** 1.

5. Definujte funkciu *menu*, ktorá výzve užívateľa, aby zadal vstup z klávesnice, v prípade, že vstup je znak `s`, funkcia vykreslí tvorec pomocou `turtle`, v prípade, že vstup je znak `t`, funkcia vykreslí trojuholník pomocou `turtle`, inak funkcia vypíše 'Zadali ste neplatný vstup.'.

6. Definujte funkciu *pocet_rovnakych*, ktorá bude mať tri parametre (čísla *a,b,c*). Funkcia **vráti** číslo 3, ak sú všetky tri čísla *a,b,c* rovnaké, vráti číslo 2 ak sú dve z čísel *a,b,c* rovnaké a tretie číslo je iné, a vráti číslo 0, ak sú všetky tri čísla rôzne.

Napríklad: volanie *pocet_rovnakych(2,5,8)* **vráti** hodnotu 0; volanie *pocet_rovnakych(1,2,1)* **vráti** 2; volanie *pocet_rovnakych(10,10,10)* **vráti** 3.

7. Definujte funkciu s parametrom *n*, ktorá načíta *n* čísel z klávesnice a **vráti** počet načítaných čísel, ktoré sú deliteľné číslom 5.

Napríklad: funkcia pre vstupný parameter *n=5* načíta 5 čísel z klávesnice; ak tieto čísla sú napríklad: 5, 10, 3, 4, 0 funkcia **vráti** číslo 3 (lebo 5, 10 a 0 sú deliteľné piatimi).

Funkcia pre vstupný parameter *n=4* načíta 4 čísla z klávesnice; ak tieto čísla sú napríklad: 1, 2, 3, 4, funkcia **vráti** číslo 0, pretože žiadne z čísel 1,2,3,4 nie je deliteľné 5.

8. Definujte funkciu s parametrom *n*, ktorá načíta *n* čísel z klávesnice a **vráti** súčet načítaných čísel.

Napríklad: funkcia pre vstupný parameter *n=5* načíta 5 čísel z klávesnice; ak tieto čísla sú napríklad: 10, 20, 30, 40, 50 funkcia **vráti** číslo 150, lebo $10+20+30+40+50 = 150$.

Funkcia pre vstupný parameter *n=4* načíta 4 čísla z klávesnice; ak tieto čísla sú napríklad: 0,-3,2, -5 funkcia **vráti** číslo -6, lebo $0+(-3)+2+(-5) = -6$.

9. Definujte funkciu s parametrom n , ktorá načíta n čísel z klavesnice a **vráti** najmenšie načítané číslo.

Napríklad: funkcia pre vstupný parameter $n=4$ načíta z klavesnice 4 čísla; ak tieto čísla sú napríklad 8, 2, 5, 3, funkcia **vráti** číslo 2.

Funkcia pre vstupný parameter $n=5$ načíta z klavesnice 5 čísiel; ak tieto čísla sú napríklad 5,-2,1,2,3 funkcia **vráti** číslo -2.

10. Definujte funkciu s parametrom n , ktorá načíta n čísel z klavesnice a vráti druhé najväčšie načítané číslo. Môžete predpokladať, že všetky načítané čísla budú rôzne.

Napríklad: funkcia pre vstupný parameter $n=4$ načíta z klavesnice 4 čísla; ak tieto čísla sú napríklad 8, 2, 5, 3, funkcia **vráti** číslo 5.

Funkcia pre vstupný parameter $n=5$ načíta z klavesnice 5 čísiel; ak tieto čísla sú napríklad 5,-2,1,2,3 funkcia **vráti** číslo 3.

Funkcia pre vstupný parameter $n=3$ načíta z klavesnice 3 čísla; ak tieto čísla sú napríklad -10, -5, 0, funkcia **vráti** číslo -5.

Sekcia 2: Úlohy o prvočíslach

Poznámka: Úlohy v tejto sekcii na seba nadväzujú!

11. Definujte funkciu *delitelnost(a,d)*, ktorá pre argumenty a a d vráti hodnotu True, ak $a \% d == 0$, a hodnotu False, ak $a \% d > 0$.

Pomocou funkcie *delitelnost* vypíšte všetky delitele čísla 12. (Delitele čísla 12 sú 1, 2, 3, 4, 6, 12.)

Napríklad: volanie *delitelnost(6,3)* **vráti** hodnotu True (pretože 6 je deliteľné 3 teda $6 \% 3 == 0$), volanie *delitelnost(7,3)* **vráti** hodnotu False (pretože $7 \% 3 > 0$).

12. Číslo a väčšie ako 2 je prvočíslo, ak pre každé číslo d z množiny $\{2, 3, \dots, a-1\}$ platí, že $a \% d > 0$. Pomocou funkcie *delitelnost* definujte funkciu *test_prvociselnosti*, ktorá pre argument a vráti hodnotu True, ak a je prvočíslo, a hodnotu False, ak a nie je prvočíslo. Môžete predpokladať, že $a > 2$.

Pomocou funkcie *test_prvociselnosti* vypíšte všetky prvočísla väčšie ako 2 a menšie ako 50.

Napríklad: *test_prvociselnosti(5)* **vráti** True, volanie *test_prvociselnosti(23)* **vráti** True, volanie *test_prvociselnosti(21)* **vráti** False, volanie *test_prvociselnosti(27)* **vráti** False.

13. Upravte funkciu *test_prvociselnosti* tak, aby funkcia vrátila hodnotu True aj v prípade, že $a=2$.

Pomocou upravenej funkcie *test_prvociselnosti* vypíšte všetky prvočísla menšie ako 50. (číslo 2 je najmenšie prvočíslo)

Sekcia 3: Sachové úlohy

Poznámka: Úlohy v tejto sekcii sú prevzaté zo stránky

https://snakify.org/en/lessons/if_then_else_conditions/problems/

POZOR! Všetky úlohy v tejto sekcii majú krátke riešenie iba na pár riadkov! Snajte sa na toto riešenie prísť!

14. Definujte funkciu *pohyb_veze*. Funkcia bude mať štyri parametre: x_1, y_1, x_2, y_2 . Každý z parametrov môže nadobúdať hodnoty od 1 po 8. Funkcia vráti hodnotu True, ak sa sachová veža

dokaze na jeden tah dostat z pozicie (x1,y1) na sachovnici do pozicie (x2,y2). Inak vrati funkcia hodnotu False.

Pomocka: pri rozmyslani nad touto ulohou vam moze pomoct obrazok na:

https://snakify.org/en/lessons/if_then_else_conditions/problems/rook_move/

Napriklad: volanie *pohyb_veze(3,3,3,7)* **vrati** True, volanie *pohyb_veze(3,3,4,7)* **vrati** False.

15. Definujte funkciu *rovnaka_farba*. Funkcia bude mat styri parametre: x1,y1,x2,y2. Kazdy z parametrov moze nadobudat hodnoty od 1 po 8. Funkcia vrati hodnotu True, ak pozicia (x1,y1) na sachovnici ma rovnaku farbu ako pozicia (x2,y2). Inak vrati funkcia hodnotu False.

Pomocka: pri rozmyslani nad touto ulohou vam moze pomoct obrazok na:

https://snakify.org/en/lessons/if_then_else_conditions/problems/chess_board/

Napriklad: volanie *rovnaka_farba(4,2,3,7)* **vrati** True, volanie *rovnaka_farba(4,2,7,6)* **vrati** False.

16. Definujte funkciu *pohyb_krala*. Funkcia bude mat styri parametre: x1,y1,x2,y2. Kazdy z parametrov moze nadobudat hodnoty od 1 po 8. Funkcia vrati hodnotu True, ak sa sachovy kral dokaze na jeden tah dostat z pozicie (x1,y1) na sachovnici do pozicie (x2,y2). Inak vrati funkcia hodnotu False.

Pomocka: pri rozmyslani nad touto ulohou vam moze pomoct obrazok na:

https://snakify.org/en/lessons/if_then_else_conditions/problems/king_move/

Napriklad: volanie *pohyb_krala(3,3,4,3)* **vrati** True, volanie *pohyb_krala(3,3,4,5)* **vrati** False.

17. Definujte funkciu *pohyb_strelca*. Funkcia bude mat styri parametre: x1,y1,x2,y2. Kazdy z parametrov moze nadobudat hodnoty od 1 po 8. Funkcia vrati hodnotu True, ak sa sachovy strelec dokaze na jeden tah dostat z pozicie (x1,y1) na sachovnici do pozicie (x2,y2). Inak vrati funkcia hodnotu False.

Pomocka: pri rozmyslani nad touto ulohou vam moze pomoct obrazok na:

https://snakify.org/en/lessons/if_then_else_conditions/problems/bishop_move/

Napriklad: volanie *pohyb_strelca(3,3,1,5)* **vrati** True, volanie *pohyb_strelca(3,3,1,4)* **vrati** False.

18. Definujte funkciu *pohyb_damy*. Funkcia bude mat styri parametre: x1,y1,x2,y2. Kazdy z parametrov moze nadobudat hodnoty od 1 po 8. Funkcia vrati hodnotu True, ak sa sachova dama dokaze na jeden tah dostat z pozicie (x1,y1) na sachovnici do pozicie (x2,y2). Inak vrati funkcia hodnotu False.

Pomocka: pri rozmyslani nad touto ulohou vam moze pomoct obrazok na:

https://snakify.org/en/lessons/if_then_else_conditions/problems/queen_move/

Napriklad: volanie *pohyb_damy(3,3,8,3)* **vrati** True, volanie *pohyb_damy(3,3,5,4)* **vrati** False.

19. Definujte funkciu *pohyb_jazdca*. Funkcia bude mat styri parametre: x1,y1,x2,y2. Kazdy z parametrov moze nadobudat hodnoty od 1 po 8. Funkcia vrati hodnotu True, ak sa sachovy jazdec dokaze na jeden tah dostat z pozicie (x1,y1) na sachovnici do pozicie (x2,y2). Inak vrati funkcia hodnotu False.

Pomocka: pri rozmyslani nad touto ulohou vam moze pomoct obrazok na:

https://snakify.org/en/lessons/if_then_else_conditions/problems/knight_move/

Napriklad: volanie *pohyb_jazdca(5,4,3,3)* **vrati** True, volanie *pohyb_jazdca(5,4,6,5)* **vrati** False.