

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 pristúpiť ako k výzve!** Ak sa vám podarí ju na cvícení vyriešiť, ohlaste sa cvičiacemu, odprezentujte mu vaše riešenie a ak to budete mať dobre, dostanete jeden **bonusový bod**.
- **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 `%`)
3. Definujte funkciu *minimum_dvoch*, ktorá bude mať dva parametre (čísla *a,b*), a ktorá **vráti** minimum z týchto dvoch čísel.
4. Definujte funkciu *minimum_troch*, ktorá bude mať tri parametre (čísla *a,b,c*), a ktorá **vráti** minimum z týchto troch čísel.
5. Definujte funkciu *menu*, ktorá výzve užívateľa, aby zadal vstup z klávesnice, v prípade, že vstup je znak *s*, funkcia vykreslí štvorec 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.
7. Definujte funkciu s parametrom *n*, ktorá načíta *n* čísel z klávesnice a vráti počet načítaných čísel, ktoré sú deliteľné číslom 5.
8. Definujte funkciu s parametrom *n*, ktorá načíta *n* čísel z klávesnice a vráti súčet načítaných čísel.
9. Definujte funkciu s parametrom *n*, ktorá načíta *n* čísel z klávesnice a vráti najmenšie načítané číslo. Otestujte vašu funkciu pre prípad, že načítané čísla sú 8, 2, 5, 3.
10. **Bonusová úloha.** Definujte funkciu s parametrom *n*, ktorá načíta *n* čísel z klávesnice a vráti druhé najväčšie načítané číslo. Môžete predpokladať, že všetky načítané čísla budú rôzne.

Sekcia 2: Ulohy o prvocislach

Poznamka: Ulohy v tejto sekcii na seba nadvazuju!

11. Definujte funkciu *delitelnost*, ktora pre argumenty a a d vrati hodnotu `True`, ak $a \% d == 0$, a hodnotu `False`, ak $a \% d > 0$.

Pomocou funkcie *delitelnost* vypiste vsetky delitele cisla 12. (Delitele cisla 12 su 1, 2, 3, 4, 6, 12.)

12. Cislo a vacsie ako 2 je prvocislo, ak pre kazde cislo d z mnoziny $\{2, 3, \dots, a-1\}$ plati, ze $a \% d > 0$. Pomocou funkcie *delitelnost* definujte funkciu *test_prvociselnosti*, ktora pre argument a vrati hodnotu `True`, ak a je prvocislo, a hodnotu `False`, ak a nie je prvocislo. Mozete predpokladat, ze $a > 2$.

Pomocou funkcie *test_prvociselnosti* vypiste vsetky prvocisla vacsie ako 2 a mensie ako 50.

13. Upravte funkciu *test_prvociselnosti* tak, aby funkcia vratila hodnotu `True` aj v pripade, ze $a=2$. Pomocou upravenej funkcie *test_prvociselnosti* vypiste vsetky prvocisla mensie ako 50. (cislo 2 je najmensie prvocislo)

Sekcia 3: Sachove ulohy

Poznamka: Ulohy v tejto sekcii su prevzate zo stranky

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

POZOR! Vsetky ulohy v tejto sekcii maju kratke riesenie iba na par riadkov! Snazte sa na toto riesenie prist!

14. Definujte funkciu *pohyb_veze*. 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 veza 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 pomoc obrazok na:

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

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 pomoc obrazok na:

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

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 pomoc obrazok na:

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

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 pomoc obrazok na:

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

18. Definujte funkciu *pohyb_damy*. Funkcia bude mat styri parametre: x_1, y_1, x_2, y_2 . 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 (x_1, y_1) na sachovnici do pozicie (x_2, y_2) . Inak vrati funkcia hodnotu False.

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

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

19. Definujte funkciu *pohyb_jazdca*. Funkcia bude mat styri parametre: x_1, y_1, x_2, y_2 . 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 (x_1, y_1) na sachovnici do pozicie (x_2, y_2) . Inak vrati funkcia hodnotu False.

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

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