

F1420 - cvičenia na podmienky a rekurziu

Nasledujúce sú príklady 5.2, 5.3, 5.5, 5.6, 6.2, 6.4, 6.5 z knihy Think Python.

1

Velká Fermatova veta hovorí:

Nejestvujú celé čísla x , y a z väčšie ako nula, pre ktoré by platilo $x^n + y^n = z^n$, kde n je prirodzené číslo väčšie ako 2.

Napište funkciu `check_fermat`, ktorá vezme 4 parametre a vyskúša či Fermatova veta pre ne platí.

Podľa seriálu The Simpsons by veta **nemala platiť** pre čísla $x = 3987$, $y = 4365$, $z = 4472$, $n = 12$.

2

Ak je 1 strana trojuholníka väčšia ako súčet ostatných dvoch strán, tak tento trojuholník nenakreslite. Inak áno. Napíšte funkciu, ktorá bere 3 dĺžky strán, a overí, či je možné zostaviť trojuholník z týchto strán.

3

Python má zabudovanú funkciu `abs`, ktorá vráti absolútnu hodnotu čísla.

```
abs(1.21)    # -> 1.21  
abs(-1.21)   # -> 1.21
```

Vytvorte si vlastnú verziu tejto funkcie a porovnajte s `abs`.

4

Python má zabudovanú funkciu `round`, ktorá zaokrúhli číslo na najbližšie celé číslo (toto nie je celá pravda, ale budeme sa tváriť, že je to tak, inak by to bolo na dlho 1 2 3).

```
round(1.50001)  # -> 2
```

Ako by ste túto funkciu napísali pomocou funkcie `int`? Vyskúšajte a porovnajte s funkciou `round` pre nasledujúce hodnoty: 1, 1.1, 1.499, 1.51 -1.1, -1.51.

(`int` vyberá celé číslo bližšie k nule.)

5

Prečítajte si nasledujúci kód a snažte sa zistiť ako bude výsledný obrázok vyzerat:

```
import turtle

def draw(t, length, n):
    if n == 0:
        return
    angle = 50
    t.forward(length * n)
    t.left(angle)
    draw(t, length, n-1)
    t.right(2 * angle)
    draw(t, length, n-1)
    t.left(angle)
    t.backward(length * n)

stromozrava_korytnacka = turtle.Turtle()
stromozrava_korytnacka.speed(0)

n = 5
l = 50 / n

draw(stromozrava_korytnacka, l, n)

# input()      # pocka na to az stacite Enter (nezavrie okno)
```

Potom, vyskúšajte funkciu (pre viac n) a vysvetlite ju.

6

- Příklad 5.6 z Think Python, část 1 a 2.
- Příklady 6.4 a 6.5 z Think Python.

7

Príklad 6.2 z Think Python. Je možné, aby pre nejaké hodnoty m a n funkcia nikdy (akože fakt nikdy - bez ohľadu na život vášho počítača alebo vesmíru) nedobehla?

Viac o Ackermannovej funkcii v [tomto videu](#).