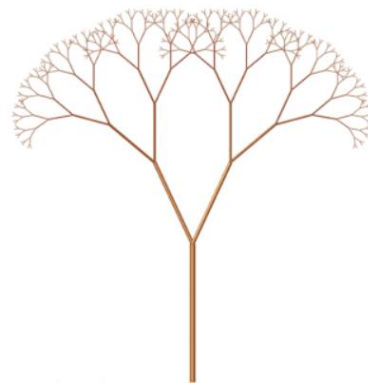


# PRACOVNÝ LIST

## PROGRAMOVANIE – FUNKCIE S PARAMETRAMI PRE KRESLENIE

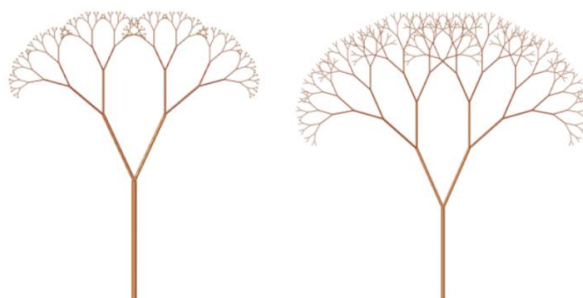
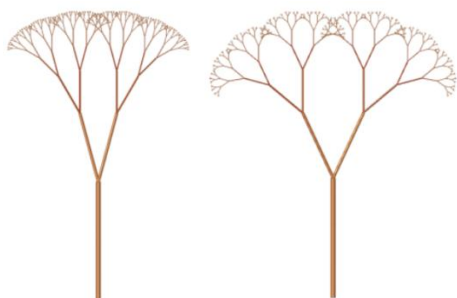
### ÚLOHA 1

Preskúmajte strom na obrázku a pokúste sa nájsť nápad, ako by sa dal nakresliť pomocou nejakého jednoduchého geometrického vzoru (načrtnite ho).



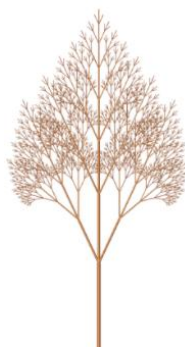
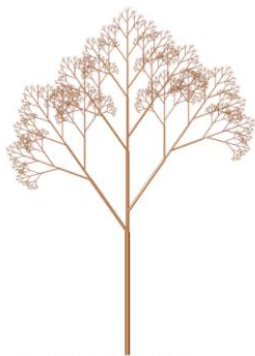
### ÚLOHA 2

Preskúmajte dvojice stromov a zistite, v čom sa líšia pri Vašom pôvodne navrhnutom geometrickom vzore:



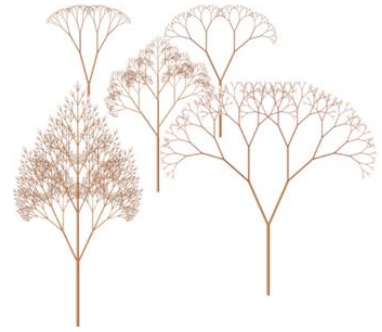
### ÚLOHA 3

Preskúmajte nasledujúce dva stromy na obrázku a načrtnite k nim zodpovedajúci geometrický vzor (porovnajte ho so vzorom z úlohy 1).



## ÚLOHA 4

Ako by teda mohli vyzeráť parametre funkcie **strom**, ktorou by sme dokázali vykresľovať rôzne stromy?

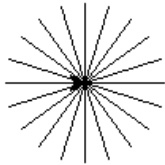


```
def strom (
    ,
    ,
    ,
    )
```

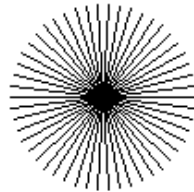
Ako by sa dali ešte pomocou ďalších parametrov meniť/vylepšovať vykresľované stromy?

## ÚLOHA 5

- a) Navrhните funkciu **luce(dĺzka)**, ktorá bude vykresľovať rôzne veľké slnko podľa tohto vzoru:

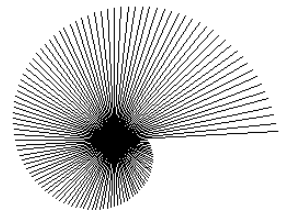


- b) Vylepšite svoju funkciu na funkciu **luce(dĺzka, pocet)** tak, aby vedela kresliť slnká aj s rôznym počtom lúčov, teda napr. :



```
def luce(dlzka, pocet):
```

Pomocou neznámej funkcie **divne\_luce** sme nakreslili nasledujúci obrázok:  
V čom sa naša neznáma funkcia líši od Vašej funkcie **luce**?

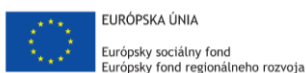


## ÚLOHA 6

- a) Vytvorte súbor **kruhy.py** s nasledovným krátkym kódom:

```
import turtle
pero=turtle.Turtle()
pero.circle(50)
```

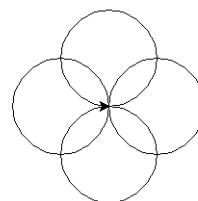
Vyskúšajte ho a zistite, ako funguje príkaz **circle** . Čo predstavuje jeho parameter (t.j. hodnota 50)?



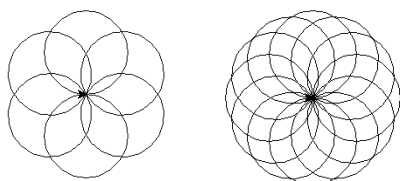
Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu  
a Európskeho fondu regionálneho rozvoja v rámci Operačného programu Ľudské zdroje

[www.minedu.sk](http://www.minedu.sk) [www.employment.gov.sk/sk/esf/](http://www.employment.gov.sk/sk/esf/) [www.itakademia.sk](http://www.itakademia.sk)

b) Vytvorte funkciu **ruzica**, ktorá nakreslí nasledovnú ružicu:



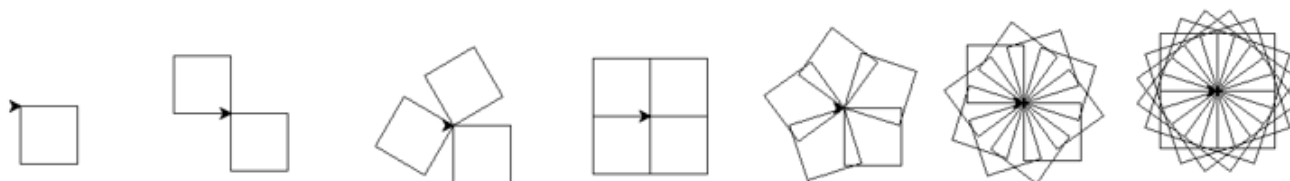
c) Upravte funkciu **ruzica** tak, aby ste pomocou nej vykreslili rôzne ružice, napr.:



```
def ruzica(velkost, pocet_lupenov):
```

## ÚLOHA 7

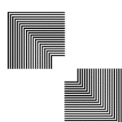
Vytvorte súbor **stvorec.py** s funkciou **stvorec** na vykresľovanie rôzne veľkých štvorcov a funkciou **obrazec**, ktorá bude pomocou volania funkcie **stvorec** vykresľovať nasledovné obrazce:



```
def obrazec(velkost, pocet):
```

## ÚLOHA 8 (POVINNÁ)

a) Upravte funkciu **obkladacka** v súbore **obklad.py** na funkciu **novy\_vzor(velkost, farba)** tak, aby vykresľovala na obkladacku symetrický vzor danej veľkosti a farby podľa požiadaviek zákazníka, napr.:

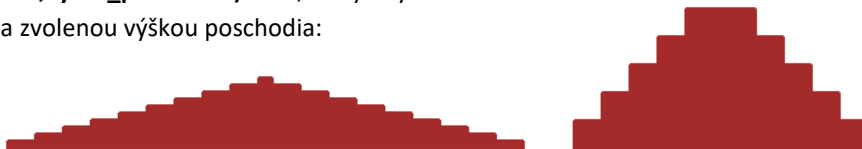


b) Doplňte do programu funkciu **dekoracny\_pas(pocet, velkost, farba)**, ktorá vykreslí stanovený počet obkladáčiek požadovanej veľkosti a farby v dekoračnom páse do kúpeľne, napr.:



## ÚLOHA 9 (POVINNÁ)

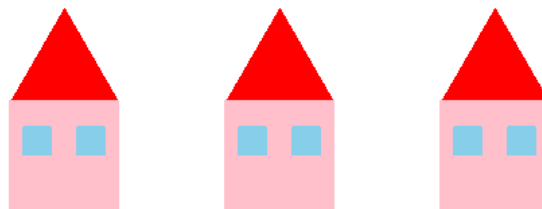
Upravte svoju funkciu **pyramida** v súbore **pyramidy.py** z minulej hodiny (úloha 11 v pracovnom liste, príp. použite priložený rovnomenný súbor z priečinku s pracovnými súbormi) na **pyramida(pocet\_poschodi,rozmer\_vrcholu,vyska\_poschodia)** tak, aby vykresľovala pyramídu s určeným rozmerom vrcholu a zvolenou výškou poschodia:



Poznámka: Pokiaľ pracujete s priloženým pracovným súborom, na vyriešenie úlohy budete potrebovať modifikovať aj funkciu **poschodie**.

## ÚLOHA 10 (DOBROVOĽNÁ)

- a) V súbore **dom.py** z minulej hodiny (úloha 10 v pracovnom liste, príp. použite priložený rovnomenný súbor z priečinku s pracovnými súbormi) vytvorte funkciu **ulica(velkost,pocet,vzdialenost,farba)** na vykreslenie požadovaného počtu rovnako veľkých domov s určeným rozstupom medzi nimi a farbou omietky:



- b) Dopĺňte do programu na záver aj vykreslenie cesty alebo chodníka spájajúceho všetky nakresleného domy:



## ÚLOHA 11 (SAMOHODNOTIACI TEST)

Načrtnite, aký obrazec vykreslí nasledujúci program:

```
def utvar1(velkost, farba):  
    pero.color(farba)  
    pero.begin_fill()  
    for i in range(4):  
        pero.forward(velkost)  
        pero.right(90)  
    pero.end_fill()  
  
def utvar2(pocet, rozmer, farba):  
    for i in range(pocet):  
        pero.forward(rozmer)  
        utvar1(rozmer/2, farba)  
        pero.backward(rozmer)  
        pero.right(360/pocet)  
  
utvar2(6, 60, "green")
```