

Lista 3

Każde zadanie należy umieścić w osobnym pliku `imie.nazwisko.XX.YY.py` gdzie `XX` to numer listy a `YY` numer zadania. Wyjątkiem jest zadanie 6 - każdy podpunkt w osobnym pliku.

Zadanie 1 (1 pkt)

Wykorzystaj listę składaną (*list comprehension*), aby stworzyć sekwencję kwadratów liczb naturalnych mniejszych od 100. Następnie wydrukuj na ekranie:

```
1 -> 1
2 -> 4
3 -> 9
.
.
.
```

Zadanie 2 (1 pkt)

Przeanalizuj poniższy kod:

```
#!/usr/bin/env python

i = 0

# drukujemy wszystkie liczby parzyste mniejsze od 10
while i < 10:
    if i % 2: # reszta z dzielenia != 0 -> True
        continue # pomiń liczby nieparzyste
    else:
        print(i) # drukuj liczby parzyste

    i += 1 # zwiększ i o jeden
```

Czy skrypt będzie działał zgodnie z założeniami? Jeśli nie, to napraw go.

Zadanie 3 (2 pkt)

Uzupełnij skrypt o brakujące fragmenty:

```
#!/usr/bin/env python

# lista zakupów
grocery = ['jajka', 'mleko', 'chleb', 'maslo', 'piwo']
# ilość sztuk
n_items = []
# zakazane produkty
prohibited = ['wódka', 'piwo', 'wino']

# w pętli pytamy użytkownika, ile sztuk danego produktu chce kupić
for product in grocery:
    # wydrukuj na ekranie komunikat: "Produkt [nazwa produktu]: sztuk = "
    # pobierz liczbę od użytkownika i zapisz w n_items
    # pomiń produkty zakazane (i automatycznie przyjmij ilość = 0)

# drukujemy listę zakupów

print("{:-^50}".format("Lista zakupów"), end="\n\n")

# w pętli wydrukuj: [lp]. [nazwa produktu]: [ilość]
# czyli np.: 1. jajka: 5 itd.
```

Zadanie 4 (3 pkt)

Napisz skrypt, który:

- losuje liczbę całkowitą mniejszą od 100 (`help(random.randint)`)
 - pyta użytkownika o odgadnięcie liczby
 - informuje użytkownika, czy podana przez niego liczba jest:
 - dużo mniejsza (różnica > 50)
 - mniejsza (różnica > 10)
 - trochę mniejsza
 - trochę większa
 - większa (różnica > 10)
 - dużo większa (różnica > 50)
 - program się kończy, gdy użytkownik odgadnie wylosowaną liczbę
-

Zadanie 5 (4 pkt)

Napisz program, który pobierze od użytkownika liczbę naturalną N . Program powinien pytać do skutku, aż zostaną podane prawidłowe dane.

Następnie, dla wszystkich liczb naturalnych n niemniejszych od N wydrukowane zostaną wyrazy ciągu wg poniższych reguł:

- drukuj Syk, jeśli n jest podzielne przez 3
- drukuj Bzyk, jeśli n jest podzielne przez 5
- drukuj SykBzyk, jeśli n jest podzielne przez 3 i 5
- drukuj n w pozostałych przypadkach

Przykład:

1, 2, Syk, 4, Bzyk, Syk, 7, 8, Syk, Bzyk, 11, Syk, 13, 14, SykBzyk, 16, 17, ...

Za zadanie do zdobycia 3 pkt.

Dodatkowy punkt za rozwiązanie bez użycia instrukcji warunkowej `if`.
Wskazówka: możliwe jest wykorzystanie operacji logicznych do wyznaczenia argumentu funkcji `print`, np.

```
>>> print("" or "Pusta sekwencja jest False, więc drukuję to co po or.")  
Pusta sekwencja jest False, więc drukuję to co po or.
```

Zadanie 6 (4 pkt)

Poniższy skrypt narysuje kwadrat:

```
#!/usr/bin/env python  
  
import turtle  
  
length = eval(input("Podaj długość boku: "))  
n_sides = 4 # ilość boków  
  
turtle.speed(20) # ustal prędkość żółwia  
  
# powtórz n_sides razy  
for i in range(n_sides):  
    turtle.forward(length) # narysuj linię o danej długości  
    turtle.right(90)      # obróć się w prawo o dany kąt  
  
turtle.mainloop() # nie zamykaj okna po narysowaniu  
  
• zmodyfikuj go tak, aby narysował trójkąt równoboczny  
• zmodyfikuj go tak, aby narysował sześciokąt foremny
```

- zmodyfikuj go tak, aby narysował wielokąt foremny, którego liczba boków podana jest przez użytkownika
- zmodyfikuj go tak, aby wielokąt rysowany był N razy (N podane przez użytkownika); każdy kolejny obrócony o odpowiedni kąt (aż do wykonania pełnego kąta - jak na przykładzie poniżej)

Każdy podpunkt za 1 pkt.

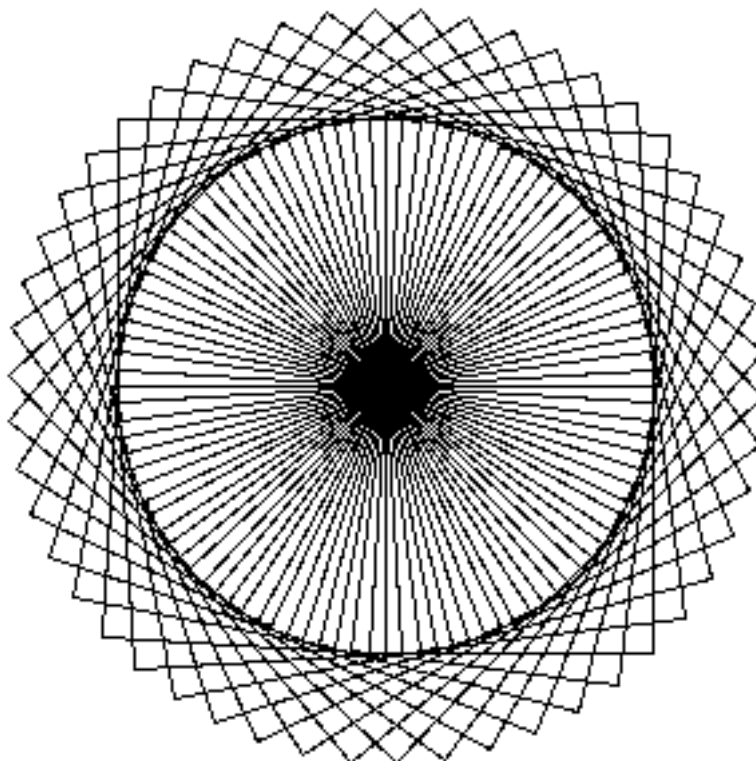


Figure 1: przykład dla 50 kwadratów