

Python (8) - funkcje

(1) Napisz funkcję *trojka*, która dla n wyświetla w 1 rzędzie n , n^2 oraz n^3 .

Korzystając z funkcji wyświetl takie trójki dla liczb od 1 do 10.

(2) Napisz funkcje *min*, *max* z dwoma parametrami zwracającą minimum i maksimum z pary liczb. Użytkownik wprowadza 2 liczby i korzystając z obu funkcji wyświetlone są te liczby w kolejności: mniejsza, większa.

Zmodyfikuj funkcje *min* i *max* tak aby mogły przyjmować 1 parametr lub więcej (składnia $\min(a, *l)$, gdzie a jest pierwszym parametrem a $*l$ jest stałą listą (tuple) z resztą parametrów typu $l[0], l[1], \dots$)

Użytkownik wprowadza 4 liczby i stosując funkcje wyświetlone są najmniejsza i największa z tych liczb.

(3) Napisz funkcję *potega* (NIE stosując $**$), która ma zwracać a^b dla liczb naturalnych a i b . Wyświetl potęgi 2 i 3 od 1 do 10.

(4) Napisz funkcję *silnia* zwracającą silnię z liczby n . Wylicz $\frac{2^n n!}{n^n}$ oraz $\frac{3^n n!}{n^n}$ dla n od 1 do 10. Do czego wydają się zmierzać wyrazy tych ciągów? Przetestuj kilka podobnych ciągów dla a^n zamiast 2^n gdzie a jest pomiędzy 2 i 3 zwracając uwagę na to czy ciągi rosną czy maleją.

(5) Napisz rekurencyjnie funkcję *silnia* (czyli wewnątrz funkcji wzywamy samą funkcję z innym parametrem niż wyjściowy). Przetestuj dla coraz większych liczb, zwracając uwagę od jakiego momentu czas wykonania programu jest odczuwalny.

(6) Napisz rekurencyjnie funkcję *fibb*(n) oddającą n -ty element ciągu Fibonacciego:

$a_1 = 1, a_2 = 1$ oraz $a_n = a_{n-1} + a_{n-2}$ dla $n \geq 3$

Przy jakich n obliczenia stają się wolne? Niech $b_n = (\sqrt{5} \text{fibb}(n))^{1/n}$. Oblicz b_n dla n od 20 do 30. Co wychodzi? Wyświetl $(2 * b_n - 1)^2$ i wywnioskuj do czego dąży b_n . (liczba ta nazywa się złotym środkiem i ozanaczana jest przez ϕ).

Wywnioskuj czemu w przybliżeniu jest równe *fibb*(n) dla dużych n .

(7) (Dodatkowe) Napisz funkcję *scyfry*(n) która oddaje sumę cyfr liczby n :

np. dla $n = 34215$ funkcja zwraca 15 bo $3 + 4 + 2 + 1 + 5 = 15$.

Przetestuj, że funkcja działa.

Wyświetl wszystkie liczby z zakresu [1000,10000] których suma cyfr jest równa 33.

Oblicz w tym samym zakresie ilość liczb o sumie cyfr 1,2,3, ..., 36. Wyświetl wynik w postaci:

2

1 : 1

2 : 4

3 : 10

....