

# Lista 3

## Funkcje wyższych rzędów

W poniższych zadaniach **wolno** wykorzystywać funkcje wbudowane obliczające **długość listy**, **odwracające listę** oraz **łączące dwie listy**, o ile nie ma to wpływu na **złożoność obliczeniową**.

Każde zadanie, poza implementacją funkcji, musi posiadać **kompletny zestaw testów**.

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładzie nr 3.

- 1) Napisać operator `!?` przyjmujący listę liczb rzeczywistych `a` i zwracający funkcję argumentu `x`:

$$x \rightarrow a_0 + a_1x + a_2x^2 + a_3x^3 + \dots$$

Dla listy pustej zwracać wartość 0.0. (OCaml) (25pkt.)

- 2) Napisać funkcję *composed* generującą listę wszystkich liczb złożonych z przedziału `[1, n]`. Wykorzystać jedynie wyrażenia `for` oraz `List.range` tzn. nie można korzystać z rekurencji, ani funkcjonałów. (Scala) (25pkt.)

Wskazówka: Aby przekonwertować liczbę rzeczywistą na całkowitoliczbową w Scali należy wywołać metodę `.toInt` np. `2.57.toInt == 2`