

Lista 3

Funkcje wyższych rzędów

W poniższych zadaniach **nie wolno** wykorzystywać funkcji wbudowanych obliczających **długość listy**, **odwracających listę** oraz **łączących dwie listy**.

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

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

- 1) Napisać funkcję *revNComp* przyjmującą funkcję f i liczbę elementów n , i zwracającą funkcję argumentu x postaci:

$$x \rightarrow [f(f(\dots f(f(x))\dots)); f(\dots f(f(x))\dots); \dots; f(f(x)); f(x); x]$$

Jeżeli $n < 0$ zwrócić listę pustą. (OCaml i Scala) (6pkt.)

- 2) Napisać funkcję *area* przyjmującą parę liczb (a, b) , funkcję f oraz liczbę podziałów n ; obliczającą pole powierzchni pod wykresem funkcji f na przedziale $[a, b]$. Implementację oprzyj w całości na znanych Ci funkcjonalach dla list (wyjątek dla języka OCaml, gdzie wygenerowanie ciągu liczb wymaga użycia rekurencji). Wykorzystaj definicję całki oznaczonej. (OCaml i Scala) (14pkt.)