

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 operator >=< przyjmujący listę l oraz funkcję f i zwracającą listę postaci:

$[f\ x1\ x2; f\ x3\ x4; f\ x5\ x6; \dots]$

Jeżeli lista zawiera niesparowany element zwrócić listę pustą. Kolejność elementów ma znaczenie. Operator zdefiniować tak, aby można było używać go w „łańcuchu” tzn. np. $l\ \text{>=<}\ f\ \text{>=<}\ g\ \text{>=<}\ h$, gdzie l – lista; f, g, h – funkcje.

Wykorzystując operator >=< napisać funkcję *collapse*, która wykonuje operację >=< do czasu, aż lista nie będzie jednoelementowa lub, dla listy pustej, pusta. (OCaml) (20pkt.)

- 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). Jako metodę obliczania wartości całki oznaczonej zastosuj metodę Simpsona¹. Przy testach sprawdź podstawowe własności całek oznaczonych. (OCaml i Scala) (30pkt.)

¹ https://eduinf.waw.pl/inf/alg/004_int/0004.php