

# Lista 1

## Wstęp do programowania funkcyjnego

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 1.

- 1) Napisać funkcję *mirror4* odwracającą kolejność dwóch pierwszych oraz dwóch ostatnich elementów podanej krotki czteroelementowej. Np. (1, 2, 3, 4) -> (2, 1, 4, 3) (OCaml i Scala) (10pkt.)
- 2) Napisać funkcję *remove* przyjmującą listę, oraz pozycję, z której ma zostać usunięty element. Jeśli pozycja jest poza zakresem, należy zwrócić oryginalną listę. (OCaml i Scala) (10pkt.)
- 3) Zegarmistrz naprawia stary zegar. W jego mechanizmie znajdują się koła zębate o różnej liczbie zębów tworzące przekładnię. Napisz funkcję *rotations* przyjmującą listę liczb zębów kolejnych kół w ciągu i zwracającą liczbę obrotów na sekundę ostatniego z kół. Znak zwracanej liczby ma oznaczać, czy koło obraca się w tym samym (+), czy w przeciwnym (-) kierunku, co koło pierwsze. Przyjmij, że pierwsze koło obraca się z prędkością 1 obr/s oraz, że liczba zębów koła nie może być ujemna. (OCaml i Scala) (30pkt.)