

Lista 2

Postać zwinięta i rozwinięta. Rekurencja ogonowa. Mechanizm dopasowania wzorca

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 2.

- 1) Napisać funkcję *indexSwap* zwracającą listę z zamienionymi miejscami elementami o indeksach a i b. Jeśli dany indeks nie mieści się w zakresie listy – zwrócić listę oryginalną. Funkcja ma być napisana w taki sposób, by za jej pomocą można było wygenerować wiele funkcji zamieniających elementy między danymi pozycjami np. *indexSwap15*, która zamienia elementy o indeksach 1 i 5. Wykorzystać mechanizm dopasowania wzorca, ale **bez wyrażenia if**. (OCaml i Scala) (30pkt.)
- 2) Napisać funkcje rozdzielające daną listę na trzy równe listy: (OCaml i Scala)
 - a. *split3Rec* bazującą na zwykłej rekurencji, (10pkt.)
 - b. *split3Tail* bazującą na rekurencji ogonowej. (10pkt.)Nadmiarowe elementy list należy odrzucać.

Pamiętaj, że funkcje mają być zgodne ze specyfikacją tj. nie mogą przyjmować dodatkowych argumentów.