

# 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 *find* wyszukującą pozycje danego elementu na liście. Ma być zwracana lista pozycji elementu. Funkcja ma być napisana w taki sposób, by za jej pomocą można było wygenerować wiele funkcji szukających wartości dla konkretnej listy np. *find123*, która szuka danego elementu (argumentu) na liście [1; 2; 3]. Wykorzystać mechanizm dopasowania wzorca, ale **bez** wyrażenia **if-else**. (OCaml i Scala) (18pkt.)
- 2) Napisać funkcje rozdzielające daną listę na dwie równe listy: (OCaml i Scala)
  - a. *split2Rec* bazującą na zwykłej rekurencji, (16pkt.)
  - b. *split2Tail* bazującą na rekurencji ogonowej. (16pkt.)

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