

WAŻNE1: Do każdego zadania przygotuj 3 testy sprawdzające poprawność działania!

WAŻNE2: Wolno używać tylko funkcji bibliotecznych o złożoności obliczeniowej  $O(1)$ .

Użyj następującej definicji listy oraz listy leniwej:

```
type 'a nlist = Koniec | Element of 'a * ('a nlist);;
```

```
type 'a llist = LKoniec | LElement of 'a * (unit -> 'a llist);;
```

Pomocne funkcje z wykładu: `lfrom`, `ltake`

WAŻNE3: Wszystkie funkcje napisz przy pomocy zwykłej rekurencji!!

1) Zdefiniuj funkcję "podziel" oraz "lpodziel" dzielącą listę leniwą na dwie listy leniwe.

W pierwszej liście mają znaleźć się elementy o indeksach nieparzystych a w drugiej o parzystych.

Przykład:

`[5;6;3;2;1] -> [5;3;1]` oraz `[6;2]`

Wyniki oczywiście powinny być zapisane w postaci list zadanej reprezentacji!

Punkty: 3 (język Ocaml).

2) Zdefiniuj funkcję "ldziałanie" przyjmującą dwie listy leniwe i wykonującą podane działanie na elementach list. Wynikiem jest lista leniwa.

Użyj strumieni przedstawionych na wykładzie 5, strona 16.

Przykład:

`[1;2;3]`, `[2;3;4;5]` oraz `+` daje `[3;5;7;5]`

Wyniki oczywiście powinny być zapisane w postaci leniwej!

Punkty: 4 (język Scala).