

# Operacje na listach

## zadania

# Zadanie 1

Napisać używając `map` i `filter` funkcję

```
maleLitery :: [String] -> [String]
```

przyjmującą listę napisów i zwracającą listę złożoną tylko i wyłącznie z małych liter alfabetu oryginalnych napisów. Na przykład na liście `["TeSt", "TEST", ""]` poprawnym wynikiem będzie `["et", "", ""]`.

Wskazówka: funkcja `elem` oraz tworzenie list.

## Zadanie 2

Napisać używając `map` i `filter` funkcję

```
dlugoscPalindromow :: [String] -> Int
```

przyjmującą listę napisów i zwracającą sumę długości tych z nich, które są palindromami.

Wskazówka: funkcja `sum` i `reverse`.

## Zadanie 3

Napisać używając `iterate` funkcję

```
fib :: (Integer,Integer) -> [(Integer,Integer)]
```

przyjmującą parę liczb i zwracającą nieskończoną listę, której kolejny element  $(x', y')$  powstaje z poprzedniego  $(x, y)$  w taki sposób, że  $x'$  jest równy  $y$ , a  $y'$  jest sumą  $x$  i  $y$ . Używając tej funkcji wygenerować nieskończoną listę zawierającą kolejne elementy ciągu Fibonacciego (od 0,1,1,...).

## Zadanie 4

Zdefiniować funkcję

```
dlugosc :: [a] -> Int
```

działającą jak `length`, czyli zwracającą długość listy, używając funkcji `map` i `sum` (i nie używając `length`).

## Zadanie 5

Napisać funkcję

```
slovaDlugosci :: Char -> Char -> Integer -> [String]
```

przyjmującą dwa znaki oraz liczbę i zwracającą wszystkie słowa podanej długości, których literami są podane znaki. Przykładowo dla 'a' 'b' 2 wynikiem powinno być ["aa", "ab", "ba", "bb"]. Słowa nie muszą być podane w tej kolejności, ważne, żeby były wszystkie.

## Zadanie 6

Zaimplementować algorytm QuickSort o sygnaturze

```
quickSort :: (Ord a) => [a] -> [a]
```

## Zadanie 7

Używając `zipWith` zaimplementować `map` i `zip`.

Wskazówka: Jeśli listy nie są równej długości, elementy dłuższej "bez pary" z elementem listy krótszej są pomijane.



## Zadanie 8

Zaimplementować funkcję o sygnaturze

```
eratosthenes :: Integer -> [Integer]
```

przyjmującą liczbę  $n$  i zwracającą listę liczb pierwszych nie większych niż  $n$  obliczoną metodą Sita Eratostenesa.