

# Functional Programming - Lab Class Exercises 2

Frank C Langbein  
frank@langbein.org

Version 1.4.0

## 1. Reverse

- Write a function `rev :: [a] -> [a]` that reverses the order in a list.

## 2. RemoveAt

- Write a function `removeAt :: Int -> [a] -> (a, [a])` that removes the n-th element of a given list and reports it as a pair of that element and the remaining list.

```
> removeAt 2 "test"  
( 's', "tet" )
```

## 3. Combinations

- Write a function `combs :: Int -> [a] -> [[a]]` that generates the combinations of n distinct objects chosen from the elements of a list.

```
> combs 2 [1,2,3]  
[[1,2],[1,3],[2,3]]
```

## 4. Countdown

- Run the two different solvers of the countdown problem from the lecture and compare their execution run times.
- Create a third solver using the adjusted valid function from the lecture and again compare the execution time.

- Optional: Modify the final solver to
  - allow the use of exponentiation in expressions;
  - produce the nearest solutions if no exact solution is possible;
  - order the solutions using a suitable measure of simplicity.

## 5. Eight-Queens

- The aim is to place eight queens on a chessboard so that no two queens are attacking each other; i.e., no two queens are in the same row, the same column, or on the same diagonal.
- Represent the positions of the queens as a list of numbers 1..N, e.g. [4,2,7,3,6,8,5,1] means that the queen in the first column is in row 4, the queen in the second column is in row 2, etc.

```
> length (queens 8)
92
> head (queens 8)
[1,5,8,6,3,7,2,4]
```