

# Assignment 2

## Intermediate concepts

**Deadline: Monday, October 26, 13:55**

### 2.1 Submission instructions

1. Unzip the A2.zip folder. You should find 3 files in the `src` folder:
  - `Solutions.elm` - for exercises 2.2.1 to 2.2.4, inclusive
  - `Regex.elm` - for exercise 2.2.5
  - `FunSet.elm` - for exercise 2.2.6
2. Edit the first line of each of the source files as described in the comments.
3. Edit the source files in the `src` folder with your solutions.
4. When done, zip this A2 folder and name the zip archive with the following format:

*A2-⟨FirstName⟩-⟨LastName⟩-⟨Group⟩*

Examples of valid names:

- `A2_John.Doe_30432.zip`
- `A2_Ion.Popescu_30434.zip`
- `A2_Gigel-Dorel_Petrescu_30431.zip`

Examples of invalid names:

- `Solutions.zip`
- `A2.zip`
- `Solutii_A2_Ion.Popescu.zip`

## 2.2 Assignment exercises

### Exercise 2.2.1

1p

Write an Elm function with the signature `deck : List Card` that generates the list of cards in a 52 card deck.

Hint:

Use the definitions of `Suit`, `Face` and `Card` from the previous assignment.

Grading:

1 point for the correct solution.

### Exercise 2.2.2

1p

Write an Elm function with the signature `cardValue : Card -> List Int` that returns the possible values of a card in a blackjack game.

Hint:

<https://en.wikipedia.org/wiki/Blackjack#Rules>

Grading:

1 point for the correct solution.

### Exercise 2.2.3

1p

Write an Elm function `smallestK` that returns the  $k$  smallest numbers from a list. You can assume that  $k$  will always be strictly greater than 0.

Grading:

1 point for the correct solution.

## Exercise 2.2.4

1p

Write an Elm function with the signature `balanced : String -> Bool` which verifies the balancing of parentheses in a string.

Elm REPL

```
> balanced "(if (zero? x) max (/ 1 x))"
True : Bool
> balanced "I told him (that it's not (yet) done). (But he wasn't listening)"
True : Bool
> balanced ":-)"
False : Bool
> balanced "()("
False : Bool
```

Hint:

It's not enough to verify that a string contains the same number of opening and closing parentheses.

Grading:

1 point for the correct solution.

## Exercise 2.2.5

1p

Write an Elm function with the signature `coinChange : Int -> List Int -> Int` that counts how many different ways you can make change for an amount, given a list of coin denominations. You can assume that the amount will always be strictly greater than 0.

Elm REPL

```
> coinChange 4 [1, 2]
3 : Int
> coinChange 11 [1, 5, 10]
4 : Int
```

In the first case the possibilities are: `[[1, 1, 1, 1], [1, 1, 2], [2, 2]]` and in the second case: `[[11 * 1], [6 * 1, 5], [1, 5, 5], [1, 10]]`

Grading:

1 point for the correct solution.

## Exercise 2.2.6

2p

You are given an incomplete implementation of a very simple regex engine in the `Regex.elm` file. You will have to implement 4 functions:

- `matchLit`, which matches a character literal
- `matchOneOf`, which matches one of two patterns
- `matchMany`, which matches the pattern zero or more times
- `matchSeq`, which matches the two patterns, in order

Details about the expected behaviour of the functions are given as comments in the file.

Grading:

0.5 points for each function implementation

For this exercise you will have to implement the common set operations on sets if integers represented as functions. In the `FunSet.elm` file, you will find a type alias `FunSet` for the representation of a set as a function, the `contains` and the `singletonSet` functions. Details about the expected behaviour of the functions are given as comments in the file.

Grading:

- 0.5 points each for the `union`, `intereseect` and `diff` functions
- 0.5 points for the `fold` function
- 1 point for the `map` function