Elm Challenge Questions 1

CS 1JC3

Jack You

McMaster University

Week of Sept. 28^{th}, 29^{th}, 30^{th} 2016

Problem 1: Combinations

Given a list of n elements, write a function to generate all possible combinations of length k. This is popularly known as binomial coefficients which can be denoted by

$$\binom{n}{k}$$

For example, $C(12, 3)$ *or* $\binom{12}{3} = 220$ *possibilities*

This should be your function signature:

```
combinations : Int -> List String -> List (List String)
```

An example input:

```
> combinations 3 ["a", "b" ,"c" ,"d", "e", "f"]
> [["a","b","c"], ["a","b", "d"], ["a", "b" ,"e"], ... ]
```

Problem 2: Goldbach Conjecture

One of the oldest unsolved problems in mathematics is the Goldbach conjecture, which states that every even integer greater than 2 can be expressed as the sum of two primes. For example,

$$6 = 3 + 3$$
$$12 = 7 + 5$$

Write a function that produces the 2 prime numbers which sum up input *a*. This should be your function signature:

```
goldbach : Int -> (Int, Int)
```

An example input:

```
> goldbach 28
> (5, 23)
```

Problem 3: Gray Codes

Gray codes are an interesting reflective way of ordering the binary numeral system. It's ordering is more apparent visually, here is an example for the gray codes of bit length 2

00
01
11
10

Notice successive set of bits differ only by 1 bit. For the last challenge of this set, write a function to generate gray codes in sequence for a given input *n*. your function signature should be:

```
grayCodes : Int -> List String
```

An example input:

```
> grayCodes 3
> ["000", "001", "011", "010", "110", "111", "101", "100"]
```