CS_205 Declarative Programming 16 October 2020
Monika Seisenberger, Ulrich Berger

Lab Task 3: Defining Functions, List Comprehension

1. Basic functions

i) Create a function that takes a string as input and appends the string to itself.

ii) Create a function that takes the three coordinates of a point (x_1, x_2, x_3) in space and computes its distance to the origin $(0, 0, 0)$.

2. safetail Consider the function `safetail` that behaves in the same way as `tail`, except that `safetail` maps the empty list to the empty list, whereas `tail` gives an error in this case. Give variants `safetail1, safetail2, safetail3` using

i) a conditional expression

ii) guarded equations

iii) pattern matching

To get you started, the first solution is (using `null` to determine whether or not a list is empty.)

```
safetail1 :: <add most general type>
safetail1 xs = if null xs then [] else tail xs
```

3. List Comprehension. Using list comprehension define

i) a list with the first 10000 square numbers.

ii) a list with all squares up to 10000.

iii) a list with numbers between 200 and 800 that are not divisible by 11.

iv) a list with all numbers between 200 and 800 that are palindromes.

Hint 1) Start with easier expressions: what is

```
[x+1 | x <- [1..10]]?
```

To store your expressions, I recommend you give a name to your expression

```
ms0 :: [Int]
ms0 = [x+1 | x <- [1..10]]
```

Hint 2) In iv) you will need to transform an integer into a string. R transform a number `n` into a string use `show n`. The following is an example for using `show` (note the example will not be needed in iv):

```
appendToItself n = (show n) ++ (show n)
```

4. Perfect integers A positive integer is perfect is it equals the sum of all of its factors, excluding the number itself. Using list comprehension, define a function `perfects :: Int -> [Int]` that returns the list of all perfect numbers up to a given limit.

5. (optional) We want to find all triples (x,y,z) fulfilling the property $x^2+y^2 = z^2$ where x, y, z are positive integers/ As there are infinitely many only look at those with numbers x, y, z less than or equal to a given number `n`. So, we look for a function that maps a given number n to the list of such triples; its type is: `Int -> [(Int,Int,Int)]`