

CS3200: Programming Languages

Homework 7: Haskell functions and syntax

Required Problems

1. Write a function `blastoff :: Int -> String` that takes an integer as input and returns a countdown to a rocket launch (see examples below). If the number is 0 or less, it should return the string "Negative value, so no blastoff". For example: For example:

```
*Main> blastoff 4
"4, 3, 2, 1, Blastoff!"
*Main> blastoff 0
"Blastoff!"
*Main> blastoff 10
"10, 9, 8, 7, 6, 5, 4, 3, 2, 1, Blastoff!"
*Main> blastoff (-2)
"Negative value, so no blastoff"
```

(Hint: if you haven't read it yet, note that the function `show` converts a number to a string.)

2. Write a function `hyphenate :: [String] -> String` that takes a list of strings and returns a single string that contains the given strings in the order given, separated by "-". For example:

```
*Main> hyphenate []
""
*Main> hyphenate ["a", "b"]
"a-b"
*Main> hyphenate ["Monday", "Tuesday", "Wednesday", "Thursday" ]
"Monday-Tuesday-Wednesday-Thursday"
```

3. Write a function `multiplyMe :: Int -> [a] -> [a]` which takes as input an element of some type and a list of that same type, and returns a new list where each element is repeated `a` times. For example:

```
*Main> multiplyMe 4 []
[]
*Main> multiplyMe 1 ['a','b','c']
"abc"
*Main> multiplyMe 2 [3,1,7,5,9]
[3,3,1,1,7,7,5,5,9,9]
*Main> multiplyMe 4 ['a','b','c']
"aaaabbbbcccc"
```

4. Write a function `repeats :: Eq a => [a] -> Bool` which returns `True` if its argument contains duplicate elements. For example:

```
*Main> repeats [1,2,3,4,5]
False
*Main> repeats [1,2,3,2]
True
```