Assignment-2
Haskell Scrabble Solver, Summer 2019

Note - You may need to write helper functions for some of the problems.

1. string2int: create a function that converts from string to integer. For example, it should allow for the following computation:
>>> (string2int "25") + (string2Int "20")
>>> 40

2. You must have read about list comprehensions and the cool stuff about Haskell's support for infinite lists. Use the above functions to implement the following in Haskell:
a) Implement a function that returns an infinite list of all Fibonacci numbers
b) Implement a function that returns an infinite list of all Prime numbers
Try to write efficient functions for this problem

3. substrings: Given a string *str* and a list of strings *strlist,* return a list of all those strings which have *str* as one of their substrings. Return the strings in decreasing order of lengths. For example:
>>> substrings “abc” [“abc”, ”abcdef” , ”xv” , ”ab” , ”xabcy”]
>>> [“abcdef” , “xabcy” , “abc”]
This function might prove useful in your final project.

4. word2string : convert numbers to words:
>>>> word2string 233112234
>>>> two hundred and thirty three million, one hundred and twelve thousand, and two hundred and thirty four
>>> word2string 100
>>> one hundred
This is a lengthy problem. Try not to get frustrated :P and work in teams on this one.

5. subsets: Given a list, create a list of all the subsets of that list.
>>> subsets [1,2,3]
>>> [[ ], [1] , [2] , [3] , [1,2] , [2,3] , [3,1] , [1,2,3]]