Fall 2019 CMSE462 Bonus assignment

To be done ***individually.***

Due date: 2 December 2019.

Implement the following functions in Haskell. Test each function two times. Hand a printed report to the assistant, depicting your code, as well as the sample function calls. Each question is worth 3 points, ***to be added to your midterm grade***.

1. Define the library function **or :: [Bool] → Bool** using recursion.

2. Define a function **count_evens :: [Int] -> Int** that returns how many even numbers are in a list. For example, **count_evens [1,2,3,4,7,8,12]** should return 4.

3. The height **of** a **binary** tree is the length of the longest branch of the tree, starting from the root. Define a function **height :: Tree -> Int** that returns the height of its argument. Assume the following representation of binary trees.

```
data Tree  = Leaf Int
                       |  Node Tree Int Tree
```

4. The function **min_max :: [Int] -> Int** takes in a list of values and returns the minimum and maximum values in its argument as a pair. e.g. **min_max [3,2,6,5,1]** should return (1,6) as the result. Define the **min_max_helper :: Int -> Int -> [Int] -> (Int,Int)** function so that the **min_max** function defined below works correctly. (hint: the **min_max_helper** function should work like an accumulator)

```
min_max (h:t) = min_max_helper h h t
```

5. Define the built-in **take :: Int -> [a] -> [a]** function by using the **zip** function and list comprehension. For example, **take 3 [2,5,4,1,6]** should return [2,5,4]. (hint: think about infinite list of numbers…)

6. Define a function **series :: Int -> [Int]** in Haskell that returns an ***infinite*** list of numbers of the form [a,b,c,d,e,….] such that b-a=1, c-b=2, d-c=3, e-d=4 etc. In general, $\mathbf{n_{i+1} - n_i}$ should equal $\mathbf{n_i - n_{i-1} + 1}$. The list should start with the function's argument. For example, **take 10 (series 20)** should return [20,21,23,26,30,35,41,48,56,65]