Solve 3 problems among the 5 given problems.

Each of the 3 problems solved earns you 10 points, for a total of 30.

If you solve more than 3 problems, the highest graded 3 problems will be counted.

Solve ONE of the following 2 problems, Using Haskell.

Problem H1:
A binary tree is represented as the following data type:

```haskell
data T x = L x | B (T x) (T x) deriving (Eq, Read, Show)
```

Its postfix representation is defined by the data type:

```haskell
data Post x = Leaf x | Branch deriving (Eq, Read, Show)
```

Write a function "to_post" that converts a tree of type (T x) to its postfix form represented using the data type (Post x).

Given an input like:
```
to_post (B (L 0) (B (L 1) (L 2)))
```

it should return an output like:
```
[Leaf 0,Leaf 1,Leaf 2,Branch,Branch]
```

Given an input like:
```
to_post (B (B (L "a") (L "bc")) (B (L "apple") (L "orange")))
```

it should return:
```
[Leaf "a",Leaf "bc",Branch,Leaf "apple",Leaf "orange",Branch,Branch]
```

Problem H2:
The following Haskell program generate the infinite stream of prime numbers:

```haskell
primes = 2 : filter is_prime [3,5..] where 
is_prime p = [p]==to_primes p
to_primes n | n>1 = to_factors n p ps where (p:ps) = primes
to_factors n p ps | p*p > n = [n]
to_factors n p ps | 0==n `mod` p = p : to_factors (n `div` p)  p ps
to_factors n _ (hd:tl) = to_factors n hd tl
```

You can try it out with:
```
>take 10 primes
```

which gives:
```
[2,3,5,7,11,13,17,19,23,29]
```

Write a Haskell function that tests the following conjecture for all natural numbers smaller than 100.

The conjecture states that:
"Every even natural number greater than 2 is the sum of two prime numbers."

Your function should return True if the conjecture holds up to 100 and False otherwise.

---------------------------------------------------------------------

Solve TWO of the following 3 problems, using Python.

Problem P1:

Write a program that generates all ordered rooted binary trees with N internal nodes, that have nodes colored Red or Blue, subject to the constraint that each node has a color that's different from its sibling.

For instance, if the tree looks like this:

```
  /
 /\  
x y
```

then if x is Red, y should be Blue and if x is Blue y should be Red.

---------------------------------------------------------------------

Problem P2:

The s and k combinators are defined in Python as:

```python
s = lambda f: lambda g: lambda x: (f)(x)((g)(x))
k = lambda x: lambda y: x
```

a) Define two lambda expressions f and g that execute correctly and return an integer result when calling

```python
>>> (s)(f)(g)(42)
```

b) Apply your functions to 3 different numbers to show that for each number x:

```python
(s)(k)(f)(x) == (s)(k)(g)(x)
```

c) Explain in full detail why (s)(k)(f)(x) and (s)(k)(g)(x) give the same result for any x and for any function f and g.

---------------------------------------------------------------------

Problem P3:

A Python dictionary contains keys and values that are both hashable.

Write a Python program that checks that the set of its keys and the set of its values is in a bijection.

If that is the case, test on 3 dictionaries of your choice that if the dictionary is inverted, by using its keys as values and its values as keys and then inverted again, then the original dictionary is recovered.

Test your program with the dictionary "good":

```python
good={'one':1,'two':2,'three':3}
```

for which it should return True, and the dictionary "bad":

```python
bad={'one':1,'two':2,'three':2}
```

for which it should return False