CSE 341, Autumn 2006, Assignment 5
Miranda Part II
Due: Mon October 30, 10:00pm

5 points total (3 points each Questions 1, 2 points Question 2)

For each top-level Miranda function you define, include a type declaration. You don't need to include types for helper functions — just the top-level functions. If you get stuck figuring out the correct type, you can write the function without a type declaration, let Miranda infer it, and then add the information to your script later. However, it will be good practice to try figuring it out yourself.

You should also include a set of easy-to-run test cases. For example, for Question 1, one of your tests could be:

```
t1 = Node 7 EmptyTree (Node 4 EmptyTree EmptyTree)
tree_example1 = "t1 = " ++ show t1

tree_test1 = "tree_sum t1 = " ++ show (tree_sum t1)
```

Then define a `all_tests` function that returns a nicely-formatted string consisting of the results of running all the tests, and any test examples:

```
all_tests = lay [tree_example1, tree_test1]
```

1. The lecture notes and accompanying Miranda script `lecture.m` include a polymorphic user-defined type `tree` and a function `tree_sum`. One could also define functions `tree_product`, `tree_and`, `tree_or`, and so forth. (Note: please get the current version of these ...`tree_sum` was added recently.)

   A convenient way to do this is to define a `tree_fold` function, analogous to the `foldr` function in the Miranda standard library. Once this is done, `tree_sum` can be defined as:

   ```
   tree_sum = tree_fold (+) 0
   ```

   Write the `tree_fold` function, and using it define `tree_sum`, `tree_product`, `tree_and`, and `tree_or`.

2. A *prime pair* is a pair of prime numbers that differ by 2. The first several prime pairs are (3,5), (5,7), (11,13), (17,19), (29,31), ... It is not known whether there are an infinite number of prime pairs or not. Write a Miranda function that returns the list of all prime pairs (or at least all of the prime pairs until Miranda runs out of stack space). (You can copy and use code from `lecture.m` if you want for this question.)

Turnin: Turn in your Miranda script, including your test cases. As usual, your program should be tastefully commented, and have well-chosen tests.