Homework Assignment #2

Due Tuesday 10/22

(All assignments will be turned in at start of lecture.)

1. Write a curried sorting routine. It should take a comparison function and should return a function that sorts a list according to the given comparison function. (You may use any sorting algorithm you like. You may find the list-ref function (defined in the Scheme ref man) useful for random access to the elements of a list.)

Use your curried sort function to make a specialized version that sorts numbers in descending order and another that sorts lists in ascending order of their length. You should exploit the currying to avoid passing unnecessary arguments (i.e., you shouldn't need to mention the list argument at all).

2 a. Write a database package using Scheme. A database can be implemented as a list of pairs (i.e. an assoc list). You should provide the ability to add and delete entries in the database. You should also be able to query the database, providing a predicate and retrieving a list of entries which satisfy the predicate.

b. Modify your querying function to enable stoppable queries. To do this, the user-provided predicate should take an extra argument, which is a continuation to invoke when some condition is met, thereby exiting from the query function and not checking for any additional matches. It is the query function's responsibility to pass the predicate an appropriate continuation.

Remember that all of these database operations must be purely functional, with no side-effecting whatsoever.

3. Implement a stream of the fibonacci numbers. Recall that:

```
fib(0) = 0
fib(1) = 1
fib(n) = fib(n-1) + fib(n-2), if n>1
```

4. How would you change the meta-circular interpreter so that it implements lexical instead of dynamic scoping? Provide a description in English that is detailed enough so that someone could implement this change based on your description. (Hint: There are two small places in the code where changes are required.)