COMP 3958: Lab 3

Submit a zip file named lab3.zip containing your 2 source files: lab3q1.ml and lab3q2.ml. Your files must build without warnings or errors. Otherwise, you may not receive for it. Maximum score: 12

1. (a) Implement a function words that keeps reading words from standard input until end-of-file and returns a list of the words read. Its signature is

   ```
   val words : unit -> string list
   ```

   Test your function using I/O redirection. Note that words are separated by whitespace.

   (b) Using words, implement a program that sums integers read from standard input (until end-of-file). This version skips any word that is not an integer. The program prints the sum when it finishes.

   Name your file lab3q1.ml. Ensure that it can be built using ocamlbuild.

2. A binary search tree is usually used to store key-value pairs and we typically search for a particular key to find the corresponding value.

   Modify the binary search tree code from class to use 2 type parameters – one for the key and the other for the value. The type of the tree should be ('a, 'b) bstree, where 'a is the type of the key and 'b is the type of the value.

   The signatures of the new functions are:

   ```
   val bstree_insert : ('a, 'b) bstree -> 'a -> 'b -> ('a, 'b) bstree
   val bstree_find : ('a, 'b) bstree -> 'a -> 'b option
   val bstree_delete : ('a, 'b) bstree -> 'a -> ('a, 'b) bstree
   val bstree_of_list : ('a * 'b') list -> ('a, 'b) bstree
   ```

   - we assume that keys can be compared using the less-than operator (<);
   - for bstree_insert, if the key is already in the tree, the corresponding value is updated to the new value;
   - bstree_find basically returns the corresponding value if the key is in the tree; its return type is 'b option.

   Name your file lab3q2.ml.