COMP 3958: Lab 3

Submit a zip file named `lab3.zip` containing your source files: `question1.ml` and `kvtree.ml` (and an optional `kvtree.mli` file). Your files must build without warnings or errors. Otherwise, you may not receive credit for it. Maximum score: 16

1. (a) i. Implement from basics a function `digits` that returns the list of digits in a non-negative integer. For example, `digits(3276)` returns the list `[3; 2; 7; 6]`.

   ii. Using either `List.fold_left` or `List.fold_right`, implement a function `int_of_digits` that returns an integer from a list of its digits. For example, `int_of_digits [0; 3; 2; 7; 6]` returns 3276. Note that leading zeros in the list do not change the returned integer.

   (b) i. Using either `String.fold_left` or `String.fold_right`, implement a function `list_of_string` that returns the list of characters in a string, in the same order they occur in the string. For example, `list_of_string("hello")` returns `['h'; 'e'; 'l'; 'l'; 'o']`.

   ii. Using (i) (together with some `List` function), implement a function `is_permutation` that tests whether two strings are permutation of each other. For example,

   - `is_permutation("hello", "leolh")` and `is_permutation("hello", "ohell")` both return true;
   - `is_permutation("hello", "hell")` returns false.

   Put your code in a file named `question1.ml`.

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 variables: one for the key and the other for the value. Put your code in a file named `kvtree.ml` so that the functions will be in a module named `Kvtree` (for key-value tree). The type of the tree is `('k, 'v) t`, where `'k` is the type of the key and `'v` is the type of the value.

   The signatures of the new functions are:

```
val empty : ('k, 'v) t
val is_empty : ('k, 'v) t -> bool
val size : ('k, 'v) t -> int
val insert : 'k -> 'v -> ('k, 'v) t -> ('k, 'v) t
val find : 'k -> ('k, 'v) t -> 'v option
val delete : 'k -> ('k, 'v) t -> ('k, 'v) t
val of_list : ('k * 'v) list -> ('k, 'v) t
```

   Note: `utop` would use type variables `'a` and `'b` (instead of `'k` and `'v`) when displaying the signatures.

   - for `insert`, if the key is already in the tree, the corresponding value is updated to the new value;
   - `find` basically returns the corresponding value if the key is in the tree; it returns `None` if key is not found.

   You may need to implement additional helper functions. Optionally, you may submit a `kvtree.mli` file as well.