# COMP 3958: Lab 3

Submit a zip file named `lab3.zip` containing your 2 source files: `part1.ml` and `kvtree.ml`. Unless otherwise indicated, you are restricted to the functions in `Stdlib`. As before, your files must compile. Otherwise, you may fail to get credit. Maximum score: 14

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

   (b) 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 [3; 2; 7; 6; 8]` returns 32768. Note that leading zeros in the argument list do not change the returned integer.

   Put your code in a file named `part1.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 from class to use 2 type parameters – one for the key and the other for the value. We'll call the new tree `kvtree` (for key-value tree). Its type is `('k, 'v) kvtree`.

   Implement the following functions for `kvtree` (they are analogous to the ones for `bstree`):

```
val kvtree_empty : ('k, 'v) kvtree
val kvtree_is_empty : ('k, 'v) kvtree -> bool
val kvtree_insert : cmp:('k -> 'k -> int) -> 'k -> 'v
                    -> ('k, 'v) kvtree -> ('k, 'v) kvtree
val kvtree_find_opt : cmp:('k -> 'k -> int) -> 'k -> ('k, 'v) kvtree
                      -> 'v option
val kvtree_delete : cmp:(k' -> 'k -> int) -> 'k -> ('k, 'v) kvtree
                    -> ('k, 'v) kvtree
val kvtree_of_list : cmp:('k -> 'k -> int) -> ('k * 'v) list
                     -> ('k, 'v) kvtree
```

   Note that

   - each of the above functions has a labeled argument (`cmp`) that specifies a comparison function used to compare keys. Its purpose is similar to the `cmp` argument in `ListLabels.sort`.
   - for `kvtree_insert`, if the key is already in the tree, the corresponding value is updated to the new value;
   - the `kvtree_find_opt` function replaces the `bstree_mem` function from class; the new version needs to return the corresponding value if there is one; note that its return type is `'v option`.

   Name your file `kvtree.ml`.