

## Lab 06: Lists

Create a separate file for each question. Keep them in your “Labs” folder, with the name `liiqj` for Lab *ii*, Question *j*.

Download the headers for each function from the file `labinterface06.rkt` linked off the “Labs” page on the course Web site.

After you have completed a question (except class exercises), including creating tests for it, you can obtain feedback by submitting it and requesting a public test. Follow the instructions given in the Style Guide.

**Language level:** Beginning Student.

1. [Class exercise with lab instructor assistance] Complete the function *canadianize* from the lecture slides. This function consumes a string, *s*, and produces a new string where each “o” in *s* is replaced with “ou”.
2. Create a function *count-even-strings* that consumes a list of strings, *alist*, and produces the number of strings in *alist* that have an even length.
3. Create a function *list-pos* that consumes a nonempty list, *alist*, and an item, *item*, that is guaranteed to be in *alist* and produces the position of the first occurrence of *item* in *alist*. The first item in *alist* is at position 0.

For example `(list-pos (cons "a" (cons "b" (cons "c" (cons "b" empty)))) "b")` produces 1.

4. Create a function *switch-case* that consumes a string, *str*, and produces a string in which each lower-case letter is converted to an upper-case letter, each upper-case letter is converted to a lower-case letter, and all other characters are preserved.
5. Create a function *differences* that consumes a nonempty list of numbers, *alon*, and produces a list of differences between adjacent pairs. The function produces *empty* for a list of length 1. For example:

```
(differences (cons 25 (cons 16 (cons 9 (cons 1 (cons 4
empty)))))
```

```
=> (cons 9 (cons 7 (cons 8 (cons -3 empty))))
```

6. Create a function *next-list* that consumes a list, *alist*, and an item, *item*, and produces either the element in *alist* that appears after the **first** occurrence of *item* or the symbol `'none` if *item* is either the last element in *alist* or not in *alist*.
7. *Optional open-ended questions* Create a function that consumes a list of Boolean values, representing the binary encoding of a number (`#true` is 1 and `#false` is 0), and produces the binary encoding of a number one greater. Consider functions that double a binary number or make a binary number one smaller.