CSC108H Week 5 Lab

To earn your lab marks, you must actively participate in the lab. *You do not need to finish in the time allotted, you just need to arrive on time and work hard.*

1 Objectives

1. Practice using list methods, indexing, and slicing
2. Practice using while

2 Driver and navigator

driver: The person typing at the keyboard.

navigator: The person watching for mistakes and thinking ahead.

Log in and open Wing. **Switch driver and navigator after every step.**

3 Lists

In this section, you will write short functions or statements that involve lists. In some of the exercises, you will be expected to use list methods, so you can become familiar with the tools available to you.

1. Given a list `L` and a value `v`, write an expression that removes the **first** occurrence of `v` from `L`.
2. Write an expression that adds the string `"How are you?"` to the **front** of the list `["I am fine, thank you"]`.
3. Write an expression that turns `[2, 4, 99, 0, -3.5, 86.9, -101]` into `[99, 86.9, 4, 2, 0, -3.5, -101]`. You should use just two methods.
4. Write a function `every_third` that takes a list as a parameter and returns a new list that contains every third element of the original list starting with index 0. That is, your new list should include elements 0, 3, 6, 9, 12, ... of the original list.
5. Write a function `every_ith` that takes a list `L` and an integer `i` as parameters and returns a list consisting of every `i`*th* element of `L` starting with index 0.

Show your TA your work.

4 Nested Lists

List elements may be lists themselves (called a nested list or a list of lists):

```
pets = [["Shoji", "cat", 18], ["Hanako", "dog", 15], ["Sir Toby", "cat", 10],
       ["Sachiko", "cat", 7], ["Sasha", "dog", 3], ["Lopez", "dog", 13]]
```

We can access each element of list `pets` using its index:

```
>>> pets[0]
["Shoji", "cat", 18]
```

We can also access elements of the inner lists. For example, the third element of the second element of list `pets` (the age of the dog named "Hanako") is:

```
>>> pets[1][2]
15
```

Write the following loops and functions, and call each function to verify your work:

1. Write a loop that prints each list from list `pets` on a separate line.
2. Write a loop that prints the second element of each inner list in list `pets` on a separate line.
3. Write a loop that examines list `pets` and computes the number of dogs in the list.
4. Write a loop that examines list `pets` and computes the sum of the ages of the animals in the list. Ages are the third element of the inner lists.
5. Write a function `nested_lengths` that takes a list `L` as a parameter and returns a list of the lengths of the sublists. More formally: for each element `e` in `L`, the returned list contains a corresponding element, `c`, that represents the number of elements in `e`.

Show your TA your work.

5 `while` loops

Instead of `for` loops, use `while` loops to complete the following functions. Verify your work.

`display_list(list)`	Display the elements of the given list.
`display_list_even(list)`	Display the elements of the given list with even indices.
`display_list_reverse(list)`	Display the elements of the given list from the end of the list to the front.
`sum_elements(list)`	Sum the elements of the given list of ints, starting from the front of the list, until the total is over 100 or the end of the list is reached, and return the sum at that point (as an int).
`duplicates(list)`	Return True if the given list contains adjacent elements with the same value, and return False otherwise.