CSC120H Lab 5

1 Objectives

- Practice indexing and slicing lists
- Practice using `while` and `for`

2 List Warmup

1. Consider this code:

```
L =  [1, 2, 3, 5, 7, 11]
```

(a) Which value does `L[3]` refer to?

(a) `1` (b) `2` (c) `3` (d) `5` (e) None of the above

(b) Which error is raised by the code fragment `L[6]`?

(a) `NameError` (b) `IndexError` (c) `SyntaxError` (d) `TypeError`

(c) The slice expression `L[1:5]` produces which of the following lists?

(a) `[0, 1, 2, 3, 5, 7, 11]` (b) `[1, 2, 3, 5, 7, 11]`
(c) `[2, 3, 5, 7, 11]` (d) `[1, 2, 3, 5, 7]`
(e) `[2, 3, 5, 7]` (f) `[]`

(d) The slice expression `L[0:-1]` produces which of the following lists? Why does it produce this list?

(a) `[0, 1, 2, 3, 5, 7, 11]` (b) `[1, 2, 3, 5, 7, 11]`
(c) `[2, 3, 5, 7, 11]` (d) `[1, 2, 3, 5, 7]`
(e) `[2, 3, 5, 7]` (f) `[]`

2. Consider this code:

```
names = [’Bob’, ’Holt’, ’Zahara’, ’Amitabha’, ’Dov’, ’Maria’]
```

For the following steps, use `names` and slice notation.

(a) Write a slicing expression that produces this new list: `[’Zahara’, ’Amitabha’, ’Dov’]`
(b) Write a slicing expression that produces this new list: `[’Bob’]`
(c) Write a slicing expression that produces this new list: `[’Amitabha’, Dov’, ’Maria’]`

3. Consider this code:

```
loopy = [’little’, ’dog’, ’Fifi’]
ind = 0

while (ind < len(loopy)) and (loopy[ind] != ’dog’):
    print(loopy[ind])
    ind = ind + 1
```

What words are printed when the code above is executed? (try to do this without using Wing)

(a) `little dog` (b) `little Fifi` (c) `little` (d) None of the above

3 `while` loops and lists

Choose a **driver** and **navigator** between you and your partner.
Instructions: From the Labs page, save the file `while_list_functions.py` and write the following functions.
You **must** use `while` loops to write the functions Do *not* use `for` loops.
Note: In some of these functions, instead of returning, the expected result is that the function **prints** to the shell (their docstring descriptions will also start with the word `Print` instead of `Return` which we're used to). These functions do not return anything, and therefore should **not** contain return statements. Make sure you follow the instructions of the docstring correctly when writing your functions, and only add a `return` statement if you are asked to return something from the function.

`print_list(L):`	Print each item of `L` on a separate line. `L` is unchanged.
`print_list_even(L):`	Print the items of `L` that occur at even indices. `L` is unchanged. NOTE: Do this without using `%`, and instead changing how you increment the index in the previous function.
	Switch driver and navigator.
`print_list_reverse(L):`	Print the items of `L` from the **end** of the list to the front. `L` should not be changed, and you should not make a copy of L.
`sum_elements(L):`	Sum the items of `L`, starting from the front of list, until the total is over 100 or the end of the list is reached, and **return** the sum at that point. `L` should not be changed.

4 Nested Lists and `for` Loops

List elements may be lists themselves. When this happens it is 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]]
```

The list `pets` above has 6 inner lists. You can access each element of list `pets` using its index:

```
>>> pets[3]
["Sachiko", "cat", 7]
```

You can also access the elements of these inner lists. For example, since `pets[3]` refers to a list, you can use `pets[3][2]` to access its element at index `2`:

```
>>> pets[3][2]
7
```

This accesses the item at index `2` of the list at index `3` of list `pets` (the age of the cat named "Sachiko"), which is `7`. (Make sure you understand why this works. Ask your TA for clarification, if needed)
Instructions: From the Labs page, save the file `pets.py` and write the following `for` loops **directly under** the assignment statement for `pets` (you don't need to write functions, just write the loops one after the other in the script file).

1. Write a for loop that prints each inner list from list `pets` on a separate line.
2. Write a for loop that prints the item at index `1` (in this example, "cat" or "dog") of each inner list in list `pets` on a separate line.
   Switch driver and navigator.
3. Write a for loop that counts the number of dogs in list `pets`. Print the number of dogs after the loop.
4. Write a for loop that sums the ages of the animals in list `pets`. The last item in each inner list is the pet's age. Print out the sum of the ages after the loop.

Show your TA your work.