

### Homework # 10

Due \_\_Nov.23rd\_11:55pm\_\_

#### Submission instruction:

- Submit one .py file which contains your answers to all the questions.
- Unless specified otherwise, you should have a main program for each question.  
For example, `def main_q1():`, `def main_q2():`, etc.

#### Question 1:

a. Write a function `create_permutation(lst)` that works in-place. Given a list `lst`, this function returns a random permutation of items in `lst`.

*For example*, if `lst = [2, 5, 0, -1, 4, 9]`, one run of the function may return `[5, -1, 9, 4, 0, 2]`, another run may return `[0, -1, 2, 5, 9, 4]`.

For more detail on permutation, check this link: <https://en.wikipedia.org/wiki/Permutation>

b. Write a `main()` function that asks user for a positive integer, `n`, and prints a random permutation of the integer sequence `1, 2, ..., n`.

Your `main()` function should use the `create_permutation(lst)` function you created in a.

For example, given `n = 10`, one run of your `main()` function may print `[1, 4, 5, 2, 3, 9, 8, 7, 10, 6]`, another run may print `[4, 2, 1, 10, 8, 7, 9, 5, 3, 6]`.

#### Question 2:

Write a function `add_list( lst1, lst2)` that takes two lists of numbers of the same length and returns a list consisting of the sum of the first numbers, the sum of the second numbers, etc.

For example `add_list([1,2,3], [4,5,6])` should return the list `[5, 7, 9]` since  $1+4 == 5$ ,  $2+5 == 7$ , etc.

Your main program should prompt user for input. It should read a list of numbers, one number per line, followed by “done”, then another list of numbers, one per line, followed by done. If the lists have different lengths, your main program should print “Lists are different lengths.” If they are the same length, the main program should call `add_list` and print the resulting list, one number per line.

Note: you may define additional ‘helper’ functions in your program.

### **Question 3:**

Implement the function `create_prefix_lists(lst)` that will return a sequence of lists, each containing a prefix of `lst`. All of the lists should be collected as one big list. For example for `[2,4,6,8,10]` the function should return `[[], [2], [2,4], [2,4,6], [2,4,6,8], [2,4,6,8,10]]`

### **Question 4:**

a) Write a function, `read_menu()` that prompts the user to enter a number `n`, followed by `n` lines, each of which has the name of some food and a price. The food name and the price are separated by a ‘:’ symbol.

`read_menu()` returns a list of tuples, each of which has a string as the first component (representing a food name on the menu) and a float as the second component (representing the price).

*Example, when calling `read_menu()`, the following execution can occur:*

How many items are on the menu? 4

Enter item in the form ‘name-price’: tomato soup:3.5

Enter item in the form ‘name-price’: pea soup:3.5

Enter item in the form ‘name-price’: ham sandwich:5

Enter item in the form ‘name-price’: cheese sandwich:4.5

Your function should returned list: `[('tomato soup', 3.50), ('pea soup', 3.50), ('ham sandwich', 5.00), ('cheese sandwich', 4.50)]`

Hint: You may use the `split` function with argument “:”

*For example,*

```
>>> "tomato soup:3.5".split(":")  
['tomato soup', '3.5']
```

b) Write a function `read_customer_order()` that prompts the user to enter a sequence of items they would like to order, followed by the word "done".

`read_customer_order()` returns a list of strings, one element for each food.

*Example:*

What would you like to order?

ham sandwich

pea soup

done

Your function should return ['ham sandwich', 'pea soup']

c) Write a function `compute_price(menu_list, order_list)` that takes a list of tuples (item, price) representing the menu, and a list of strings representing the customer's order and returns the sum of the prices of the items the customer ordered.

*For example,* given the example inputs from a) and b), your function should return 8.5.

d) Write a `main()` function that calls the other functions to prompt the user to enter the menu, then prompts three customers to enter their orders and prints the total price for each of them (along with 8.5% tax and 15% tip).