

This lab is due 07/13/2014 at 11:55 p.m. (submission via Sakai)

- Please do all of the following problems in ONE file named lab5.py. This is an **INDIVIDUAL** assignment, please do all work accordingly.
- Use comments to separate your program for each problem. For questions where you should write your answers, envelop them as comments.
- For now on we will be using the Design recipe to write functions and the `assertEqual` function to run our test cases. You need to provide at least 3 test cases for each new function you write, when applicable. Please keep in mind that those elements are worth at least half of the question, so you may not want to forget them!
- The problems are worth 90 points + 10 points for attending the lab session.
- You do not need to make tests for functions that use random generation (unless they are already provided, on which case you should copy them into your file).

Problem 1: (5 points)

Write a function that takes a list(mylist) and a number(n) and returns a list of numbers that are divisible by n.

Example: Your function should return [10,5,15] for [1,10,23,5,14,15] and 5.

Problem 2: (5 points)

Write a function that takes two numbers (x,y) and returns a nested list with x rows and y columns and all values in the nested list are 0.

Example: Your function should return [[0,0],[0,0],[0,0]] for 3 and 2.

Problem 3: (5 points)

Write a function that takes a list of lists (nested list) and returns a list that consists of first elements of sub-lists. If any of the sub-lists is empty, function returns empty list.

For example your function should return [1,4,5] for [[1,8,3], [4,0,8],[5,9,2]] and [] for [[1,8,3],[],[5,9,2]]

Problem 4: (5 points)

Write a function that takes a list of numbers and returns the second maximum number in the list. Your function is guaranteed to get lists that have 2 or more elements. **Note: Do not use built-in sort function.**

Problem 5: (10 points)

Write a function that takes a list of numbers and returns the numbers and their counts in a nested list. Your function should return [[2,1],[3,2],[4,3]] for [4,4,2,3,4,3].

Problem 6: (5 points)

Write a function that receives a list of numbers and returns a list where all adjacent elements that are equal to each other have been reduced to a single element, so [6, 7, 8, 8, 9, 2, 1, 2, 2, 3] returns [6, 7, 8, 9, 2, 1, 2, 3]. You may create a new list or modify the passed in list.

Problem 7: (5 points)

Write a function that receives two lists, which are sorted in increasing order, and returns a merged list of all the elements also sorted in increasing order. You may modify the passed in lists. Your solution should work in “linear time”, which means that you are only allowed to make a single pass of both lists. **Note: You are not allowed to use built-in sort function**

Problem 8: (5 points)

Given a list of strings, return the count of the number of strings where the string length is at least 2 and the first and last chars of the string are the same.

Problem 9: (20 points)

create_Phone_Book: You are going to implement a phone book function in this problem. Your function;

- Gets no arguments.
- Asks to user enter a name and a phone number if given command is “e”.
- If entered name is already in the phone book, function creates an error message and asks user for a command (“e” or “q”). If entered name is not in the phone book, function inserts name and number to phone book and asks user for a command (“e” or “q”).
- Quits the function and returns the phone book if given command is “q”.
- Your phone book data structure should be a nested list. Here’s an example: [[“Alexis Sanchez”, “212-789-2121”], [“James Rodriguez”, “454-898-0099”], [“Anna Green”, “845-322-7898”]]

Problem 10: (5 points)

look_Up: Write a look up function for phone book that has a nested list structure (e.g. [[“Alexis Sanchez”, “212-789-2121”], [“James Rodriguez”, “454-898-0099”], [“Anna Green”, “845-322-7898”]]). Function gets a name (string) as an argument and returns the phone number.

Problem 11: (5 points)

Given a string, if its length is at least 3, add 'ing' to its end. Unless it already ends in 'ing', in which case add 'ly' instead. If the string length is less than 3, leave it unchanged. Return the resulting string.

Problem 12: (10 points)

A palindromic word is one that reads the same backwards as forwards. Hence the words hello and peel are not palindromes, but the words peep, deed and dad are palindromes.

a) Create a method called **reverse** which takes a string argument. Your method should return the reverse of the argument as a string. For example, if the argument is **FooBar** then your method should return **rabooF**.

b) Create a second method called **isPalindrome** which takes a string argument. This method should return True if the argument is a palindrome and False otherwise.

Problem 13: (5 points)

Given a string, find the first appearance of the substring 'not' and 'sad'. If the 'sad' follows the 'not', replace the whole 'not'...'sad' substring with 'happy'. Return the resulting string.

So 'Actually, he was not that sad.'

yields: 'Actually, he was happy.'