WIA2005 Algorithm Design & Analysis
Semester 2, 2018/19
LAB 1 INTRODUCTION TO PYTHON

QUESTIONS

1. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

2. Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. *Hint: how does an even / odd number react differently when divided by 2?*

3. *Take a list, say for example this one:*
   a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
   and write a program that prints out all the elements of the list that are less than 5.

4. Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)

5. Take two lists, say for example these two:
   a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
   b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
   and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

6. Ask the user for a string and print out whether this string is a palindrome or not. (A **palindrome** is a string that reads the same forwards and backwards.)

7. Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it.

8. Make a two-player Rock-Paper-Scissors game. (*Hint: Ask for player plays (using input),compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game*)

   Remember the rules:

- Rock beats scissors
- Scissors beats paper
- Paper beats rock

9. Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right.

10. Ask the user for a number and determine whether the number is prime or not.

11. Write a program that takes a list of numbers (for example, a = [5, 10, 15, 20, 25]) and makes a new list of only the first and last elements of the given list. For practice, write this code inside a function.

12. Write a program that asks the user how many Fibonnaci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate.(*Hint: The Fibonnaci seqence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, …*)

13. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.

14. Write a program (using functions!) that asks the user for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order. For example, say I type the string:
    My name is Michele
    Then I would see the string:
    Michele is name My
    shown back to me.

15. Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password. Include your run-time code in a main method.