

# Summer Projects & Workshop 2017

## Python - Assignment 1

June 1, 2017

### Basics

This section is to get you started with python (you can skip the section if you are already comfortable with them)

1. Create a program file (say 'program1.py'), write python statements to assign a variable to the string "Hello World!" and print the value of the variable. You should get comfortable with running programs from files (run by typing 'python program1.py' on terminal).
2. Try out different mathematical operators (+, -, \*, /, \*\*, %) on numbers (integers, floating-point numbers, complex numbers) and verify their functionality. Do this part on python interpreter to get familiar with it.
3. Write python statements (without using `math` library) to compute the value of:

- $\sqrt[4]{-19 + 100^2}$
- $-\frac{3 \times 6.55}{\sqrt[3]{-1200}}$

4. Run the following python commands and identify what does each command do:

```
>>> a = 9.0; b = 9; c = 4
>>> print a / c
>>> print b / c
>>> print a // c
>>> print c / b
>>> print c / a
```

Why does the output of `a / c` and `b / c` different? Similarly, the output of `c / a` and `c / b`?

5. Try out different relation operators (`==`, `!=`, `<`, `>`, `<=`, `>=`) for different data types (integers, strings, etc.). These operators work on all types, not just numbers, and return a boolean value. Make sure you understand how to use them and what they are used for.

- For a quadratic equation  $ax^2 + bx + c$ , take as input  $a$ ,  $b$  and  $c$  from the user. Write a program that computes the roots of this quadratic equation. (Note: Do handle the case when roots are complex numbers)

## Strings, Lists, Tuples, Dictionaries

- Given a string  $S = \text{"Hello World!"}$ . Try different string indexing. For example, try the following:

```
>>> S[1]
>>> S[-1]
>>> S[0:4]
>>> S[6:]
>>> S[:-1]
>>> S[:]
>>> S[::]
>>> S[::2]
>>> S[3]+S[-2:]
```

Make sure you understand what each of the following outputs before proceeding forward.

- Try the commands from previous problem for list  $L = [1, 2, 3, 4, 5, 6, 7, 8, 9]$ . This problem is to understand that the indexing of elements is similar for all sequence data types like string and list.
- Write a program to print the reverse of a string.
- Given a list of numbers  $A$ , create a new list  $B$  such that  $i^{th}$  element of list  $B$  is the sum of first  $i + 1$  elements of  $A$ .  
Example:  $A = [2, 3, 5, 7]$   
Your program should output:  $[2, 5, 10, 17]$
- Write a program that initially has a list  $A$  containing  $n$  strings and produces a dictionary with the strings of the list  $A$  as keys and the corresponding lengths of the strings as values.  
Example:  $A = [\text{'hello'}, \text{'how'}, \text{'are'}, \text{'you?'}]$   
Your program should output:  $\{\text{'hello'}:5, \text{'how'}:3, \text{'are'}:3, \text{'you?'}:4\}$
- Write a program which given an input number  $n$ , produces a list that contains tuples  $(i, i^2, i^3)$  for all numbers  $i$  from 0 to  $n$  inclusive.
- Write a program that prints the sum of all even numbers in a list.

## Conditionals, Loops

- Write a program that takes three numbers as input from the user and prints the maximum and minimum of the three numbers.

2. Write a program using a for loop that asks the user for two numbers  $a$  and  $b$  and calculates  $a^b$ .
3. Write two programs to print the following pattern using
  - (a) while loop
  - (b) for loop

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7

```
4. A word or a phrase is a palindrome if it reads the same backwards as forwards, for example: "radar", "Was it a rat I saw?" (ignoring the punctuation, capitalization and spacing). Write a program to check whether the input string is a palindrome or not.
5. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: "The quick brown fox jumps over the lazy dog". Write a program which takes as input a sentence and prints whether the sentence is a pangram or not.
6. Write a program to determine the result of two-player Rock-Paper-Scissors game. Ask the user for what the players played (using input), check if the inputs are valid (i.e. either of 'rock', 'paper' or 'scissor'), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game. If yes, repeat the game, otherwise terminate the program.  
Rules of the game: Rock beats Scissors, Scissors beats Paper, Paper beats Rock.
7. One string is an anagram of another if the second is simply a rearrangement of the first. For example, 'heart' and 'earth' are anagrams. The strings 'python' and 'typhon' are anagrams as well. For the sake of simplicity, assume that the two strings in question are made up of 26 lowercase alphabetic characters. Write a program that will take two strings and prints a boolean value whether they are anagrams.