

Python Exercises

The goal of this assignment is to get you familiarized with basic python functionalities which were discussed in the class. You will have to apply the concepts taught in class today and feel free to go through the class notebook or the other sources mentioned in the notebook to gain more clarity of the concepts.

You will have to submit a pdf copy of the notebook in slack and the submission deadline will be announced to you in class. To submit, either print the page in pdf format or you can also use the 'download as' option in the 'file' menu (you may face some errors).

Practise makes a man perfect!!! So lets get started with the exercise problems.

Write a Python program to get the Python version you are using.

```
In [1]: import sys
import math
sys.version

Out[1]: '2.7.12 (default, Dec 4 2017, 14:50:18) \n[GCC 5.4.0 20160609]'
```

Write a Python program which accepts the radius of a circle from the user and compute the area. (Use math library)

```
In [2]: radius = input('enter a value for the radius in cm : ')
area = math.pi*math.pow(radius,2)
print "the area of the circle is: {:.2f}".format(area)

enter a value for the radius in cm : 2
the area of the circle is: 12.57
```

Write a Python program to concatenate all elements in a list into a string and return it.

```
In [3]: lst1 = ["w","o","r","d"]
print "list : {}".format(lst1)
str1 = "".join(lst1)
print str1

list : ['w', 'o', 'r', 'd']
word
```

Write a Python program to calculate the area of a trapezoid.

Test Data

Height : 5

Base, first value : 5

Base, second value : 6

Expected Output: Area is : 27.5

```
In [4]: height, base1, base2 = 5, 5, 6
area = 0.5*height*(base1+base2)
print 'area of trapezoid : {:.1f}'.format(area)

area of trapezoid : 27.5
```

Write a Python program to remove and print every third number from a list of 50 natural numbers until the list becomes empty.

```
In [5]: numlst = range(1,51)
        iter = 2
        while len(numlst)>0:
            if iter<len(numlst):
                print numlst.pop(iter)
                iter += 2
            else:
                iter -= len(numlst)
```

3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
1
5
10
14
19
23
28
32
37
41
46
50
7
13
20
26
34
40
47
4
16
25
35
44
8
22
38
2
29
49
31
17
43
11

Write a Python program to print the following text, convert the string to a list and print all the words and their frequencies
"Probability is the measure of the likelihood that an event will occur. Probability is quantified as a number between 0 and 1, where, loosely speaking, 0 indicates impossibility and 1 indicates certainty. The higher the probability of an event, the more likely it is that the event will occur."

```
In [6]: str1 = "Probability is the measure of the likelihood that an event will  
        occur. Probability is quantified as a number between 0 and 1, where, loo  
        sely speaking, 0 indicates impossibility and 1 indicates certainty. The  
        higher the probability of an event, the more likely it is that the event  
        will occur."  
        lst1 = str1.lower().replace(".", "").replace(",", "").split(" ")  
        for unq_word in set(lst1):  
            print "{}:{}".format(unq_word, lst1.count(unq_word))
```

```
and:2  
probability:3  
is:3  
number:1  
an:2  
indicates:2  
as:1  
measure:1  
occur:2  
event:3  
1:2  
0:2  
between:1  
speaking:1  
higher:1  
that:2  
quantified:1  
it:1  
likelihood:1  
a:1  
of:2  
certainty:1  
loosely:1  
likely:1  
will:2  
impossibility:1  
the:6  
more:1  
where:1
```

Write a Python program to reverse the digits of a given number and add it to the original, If the sum is not a palindrome repeat this procedure.

Note: A palindrome is a word, number, or other sequence of characters which reads the same backward as forward, such as madam or racecar.

```
In [7]: num1 = 89
while True:
    print(num1)
    num1 += int(str(num1)[::-1])
    if str(num1) == str(num1)[::-1]:
        print(num1)
        break
```

```
89
187
968
1837
9218
17347
91718
173437
907808
1716517
8872688
17735476
85189247
159487405
664272356
1317544822
3602001953
7193004016
13297007933
47267087164
93445163438
176881317877
955594506548
1801200002107
8813200023188
```

Write a Python program to generate a list of squares of numbers between 1 and 20 (both nos included) and print ONLY the first and last 5 elements.

```
In [8]: numlst = [int(math.pow(x,2)) for x in range(1,21)]
print "First five square numbers: {} , Last five square numbers: {}".format(numlst[:5], numlst[-5:])
```

```
First five square numbers: [1, 4, 9, 16, 25] , Last five square numbers:
[256, 289, 324, 361, 400]
```

Write a Python program to split the following sentence based on white space and then extract the first letter of each split string.

"Jack of all trades but master of none"

```
In [9]: str1 = "Jack of all trades but master of none"
lst1 = str1.split(" ")
for x in lst1:
    print x[0]
```

```
J
o
a
t
b
m
o
n
```

Write a Python program to remove the duplicate elements in the list but preserve the order.
Eg: [2,3,5,4,2,3,8,4] should give an output [2,3,5,4,8]

```
In [10]: lst1, lst2 = [2,3,5,4,2,3,8,4], []
         for x in lst1:
             if lst2.count(x)==0:
                 lst2.append(x)
         print lst2

[2, 3, 5, 4, 8]
```

Write a Python program to solve the Fibonacci sequence using recursion.

```
In [11]: def fibonacci(length):
         if length==2:
             return [0,1]
         else:
             fiblst = fibonacci(length-1)
             return fiblst + [sum(fiblst[length-3:])]
         print fibonacci(10)

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Write a Python program to print a dictionary with keys are the names of the numbers between one and five (both included) and values are the cubes of the numbers.

Note: You should use a list to create the keys and create the dictionary using list comprehension.

```
In [12]: keys1 = ["one","two","three","four","five"]
         dict1 = {keys1[x]:(x+1)**3 for x in range(5)}
         print dict1

{'four': 64, 'three': 27, 'five': 125, 'two': 8, 'one': 1}
```

Write a python program to check whether a given number is an Armstrong number using a function.

Note : An Armstrong number is one whose sum of cubes of digits is equal to the number itself. Eg: 371

```
In [13]: num1 = 371
         cubesum = 0
         for x in str(num1):
             cubesum += int(math.pow(int(x),3))
         if num1 == cubesum:
             print "The number {} is an Armstrong number".format(num1)
         else:
             print "The number {} is not a Armstrong number".format(num1)

The number 371 is an Armstrong number
```

Write a python program to find if a number is Automorphic or not.

Note: A number is Automorphic if and only if its square ends with the same digit as itself. Eg: The number 25 has its square 625 ending with the same last digit.

```
In [14]: num1 = 76
         sqrstr = str(int(math.pow(num1,2)))
         if sqrstr[-len(str(num1)):]==str(num1):
             print "The number {} is Automorphic".format(num1)

The number 76 is Automorphic
```

Write a python program to create a matrix $X = \begin{bmatrix} 10 & 2 \\ 44 & 0.5 \end{bmatrix}$ using lists and use a function to return the sigmoid output of the matrix.

Note: The sigmoid function is given by $\frac{1}{1+e^{-x}}$ Desired output: $\begin{bmatrix} 0.9999 & 0.8807 \\ 1 & 0.6224 \end{bmatrix}$

```
In [15]: X = [[10,2],[44,0.5]]
def sigmoid(inputM):
    for x in range(len(inputM)):
        for y in range(len(X[x])):
            inputM[x][y] = 1 / (1+math.exp(-inputM[x][y]))
    return inputM
print sigmoid(X)

[[0.9999546021312976, 0.8807970779778823], [1.0, 0.6224593312018546]]
```

Create a matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ and find its transpose using

1. List comprehension
2. Zip

```
In [16]: A = [[1,2,3],[4,5,6]]
T = [[A[y][x] for y in range(len(A)) ] for x in range(len(A[0]))]
T2 = zip(A[0],A[1])
print T
print T2

[[1, 4], [2, 5], [3, 6]]
[(1, 4), (2, 5), (3, 6)]
```