Basic Python Practice Programs

Introduction

This document contains a lot of Python beginner exercises. This document also contains a solution along with the question. The question and solution are meant for dyadic purposes and optimization are possible. I encourage you to explore those opportunities once you are comfortable with the subject matter.

The difficultly level of these exercises is basic - intermediate. Intended at an audience who are new to programming or new to Python programming but have learnt the syntaxes of Python.

There is no need to solve all these in one sitting or even solve these in order. You are free to schedule the questions as you may want. I have purposefully not categorized the questions under different titles so as a particular exercise might be trying more than being slotted to a specific sub-topic.

I have personally executed the programs and made sure no errors are there. But in spite of my best efforts if you spot some mistakes, please bring them to my notice. Thanks in advance!

Exercises

1. Create a program which when run asks the user to input his/her name and then the program displays back the input with a hello message.

Solution:

```
user_name = raw_input("Please enter you name: ")
print "Hello", user_name
```

2. Create a program which when run asks the user to input a number . The program should only accept int or float.

Solution:

```
age = input("Please enter you age: ")
print "You entered: ", age
```

3. Create a program to input two numbers and display addition of two numbers

Solution:

```
print "This is a program to add two numbers"
x = input("Please enter your first number:")
y = input("Please enter your second number:")
print "-"*30
print "You entered:", x, y
print "Sum of x and y is :", x + y
```

4. Create a program to accept input from the command line and display the input provided. The program should display a usage warning if the user does not pass exactly two command line arguments

Solution:

```
import sys

if len(sys.argv) !=3 :
    print "Usage Warning!"
    print "usage:", __file__ , "arg1", "arg2"
else:
    print "You entered" ,sys.argv[1], sys.argv[2]
```

5. Create a program to accept input from the command line and display addition of the two numbers. The program should display a usage warning if the user does not pass exactly two command line arguments.

Solution:

```
import sys

if len(sys.argv) !=3 :
```

```
    print "Usage Warning!"
    print "usage:", __file__ , "arg1", "arg2"
else:
    print "You entered" ,sys.argv[1], sys.argv[2]
    print "Sum of ", sys.argv[1], sys.argv[2], "is",
int(sys.argv[1]) + int(sys.argv[2])
```

6. Create a program to accept a series of input from the user, store them in a list and then display back the provided input.

Solution:

```
more_input = True
input_list = []
while more_input:
    x = raw_input("Enter anything you wish (Enter 0 to
terminate):")
    if x == '0':
        more_input = False
    else:
        input_list.append(x)

print "There are the values you entered:"

for i in input_list:
    print i
```

7. Complete the below program to display 'hello 111'

Data:

```
x = 'Hello'
y = 111

print <fill your code here>
```

Solution:

```
x = 'Hello'
y = 111

print x + " " + str(y)
```

8. Complete the below program to display 10

Data:

```
a = '5'
b = 5

print <fill your code here>
```

Solution:

```
a = '5'
b = 5

print int(a) + b
--------------
```

9. Through a program prove that lists are mutable

Solution:

```
l = [1,2]
initial_id = id(l)
initial_len = len(l)
print "Current id of list is ", initial_id
print "Current length of list is ", initial_len

l.append(3)
post_id = id(l)
post_len = len(l)
print "id of the list post append is", post_id
print "Lenght of list post append is", post_len
if x1 == x2 and post_len > initial_len:
    print "As the ids have remained same and length of
list has increased, we can deduce that the list has
mutated itself"
    print "QED"
```

10. Complete the program to print India

Data:

```
countries = [['Germany','England','France']['Buthan',
'India', 'Sri Lanka']]
print <fill your code here>
```

Solution:

```
countries = [['Germany','England','France']['Buthan',
'India', 'Sri Lanka']]
print countries[-1][2]
```

11. Swap the values in a and b

Data:

```
a = 10
b = 20
<fill your code here>
```

Solution:

```
a = 10
b = 20
a,b = b,a
```

12. Write a program to convert a list of kilometers to a list of miles

(Hint: 1 mile = 1.6 km)

Solution:

```
kms = [1,2,3,4]
miles = []
for km in kms:
    miles.append(km * 0.62)
print miles
```

13. Given a list use DSU (Decorate-Sort-Undecorate) to sort the list on the second element of each inner list

Data:

```
l =[[1,2,3],[6,0,8],[4,7,1], [2,4,9]]

Expected output
[[6,0,8], [1,2,3], [2,4,9], [4,7,1]]
```

14. Given a list use "key" to do an inplace sorting on the second element of each inner list

```
Data:
l =[[1,2,3],[6,0,8],[4,7,1], [2,4,9]]

Expected output
[[6,0,8], [1,2,3], [2,4,9], [4,7,1]]
```

15. Write a program to accept a number and display the sum of 1 to n numbers

```
Data:
User input : 5
Output : 15

(Hint: 1+2+3+4+5 = 15)
Solution:
n = input("Enter a number:")
print sum(range(n+1))
```

16. Write a program to provide the sum of numbers in a comma separated string

```
Data:
s = '1,2,3,4,5,6,7,8,9,10'
Output for the given string : 55

Solution:
s = '1,2,3,4,5,6,7,8,9,10'
l = s.split(",")
l = map(int, l)
total = sum(l)
print total
```

17. Write a recursive program to print the factorial of a number

Data:

```
input : 5
output : 120
```

Solution:

```
def fact(n):
    if n == 1:
        return n
    else:
        return n * fact(n-1)

n = input("Enter a number")
print "Factorial of ", n, "is", fact(n)
```

Data:

```
Delhi,28000000
Bangalore,12000000
Chennai,28000000
Mumbai,67000000
```

18. Copy paste the above contents to a file. name it data.csv. Write a program to find out the total of population and average population across cities

Solution:

```
fh = open(data.csv)
total = 0
data = fh.readlines()
for line in data:
    city, population = line.split()
    total += int(population)
```

```
print "Total Population = ", total
print "Average Population = ", total/len(data)
```

---

19. Accept a string and print the reverse of the string to the console

Solution:

```
s = raw_input("Enter any string:")
print s[::-1]
```

---

20. Accept a string and check if the string is a palindrome.

Solution:

```
s = raw_input("Enter a string:")

if s == s[::-1]:
    print s ,"is a palindrome"
else:
    print s, "is not a palindrome"
```

21. Given the below string how can you remove the extra spaces in the beginning and ending of the string?

Solution:

```
s = "   hello    "
cleaned_s = s.strip()
print cleaned_s
```

---

22. Write a python program to find whether the input year is leap year or not

Solution:

```
def isleap(n):
```

```
    leap = False
    if year %4 == 0:
        if year %100 ==0:
            if year % 400 == 0:
                leap = True
        else:
            lean = True
    return leap

n = input("Enter an year greater than 1752:")

if n < 1752:
    print "Wrong input"
else:
    if isleap(n):
        print "Its a leap year"
    else:
        print "Its not a leap year"
```

23. Create a class to represent complex numbers (for eg. 5 + i6)

Data:

```
c = Complex(5,7)
d = Complex(3,-7)
print c should print 5 +i7 on the console
print d should print 3 -i7 on the console
```

Solution:

```
class Complex(object):
    def __init__(self, r, i):
        self.r = r
        self.i = i
    def __str__(self):
        if self.i < 0:
            return str(self.r) + "-i" + str(-self.i)
        else
            return str(self.r) + "+i" + str(self.i)
```

```
c = Complex(5,7)
d = Complex(3,-7)
print c
print d
```

24. Given the below list, accept user input and delete if present in the list else let the user know

Solution:

```
l = ['Bangalore','Mysore','Mumbai','New York']
print "Current list is ", l
city = raw_input("Enter a city: ")
if city in l:
    l.remove(city)
print "Current list", l
```

---

25 . Given the below list sort the list in ascending order

Solution:

```
l = ['Bangalore','Mysore','Mumbai','New York']
print "Current list", l
l.sort()
print "Post sorting", l
```

26. Write a program to accept a sentence from the user and count the number of words

Solution:

```
s = raw_input("Enter any sentence:")
print "Number of words in the sentence entered:",
len(s.split())
```

27. Write a program to accept a sentence from the user and count the number of characters

Solution:

```
s = raw_input("Enter any sentence:")
print "Number of characters in the sentence entered:",
len(s)
```

28. Write a regular expression program to accept a username from user and check for the below conditions

Data:
1. length of username should be in the range of 4 to 15
2. should contain only alphabets

Solution:

```
import re
s = raw_input("Enter a user name")
p = '^[a-zA-Z]{4,15}$'
if len(re.findall(p,s)):
    print "Passed the validation"
else:
    print "Failed validation"
```

29. Write a program to continuously accept user input and add it to a list. End the loop when user enters 0 and finally display the items in a list

Solution:

```
more_input = True
input_list = []
while more_input:
    x = raw_input("Enter anything you wish (Enter 0 to
terminate):")
    if x == '0':
        more_input = False
```

```
    else:
        input_list.append(x)

print "There are the values you entered:"

for i in input_list:
    print i
```

30 . Write a program to check if a number is a power of two

Solution:

```
n = input("Enter a number: ")
while n > 1 :
    n = n % 2
if n == 0:
    print "Entered number is a power of two"
else:
    print "Not a power of two"
```

31. Write a program to check if a given number is a prime number

Solution:

```
n = input("Enter a number greater than 1: ")
def isprime(n):
    count = 2
    def isprime
    prime = True
    while count <= n / 2:
        if n % count == 0:
            prime = False
        if not prime:
            break
        count += 1
    return prime
prime = isprime(n)
if prime:
    print "Its a prime number"
```

```
else:
    print "Its not a prime number"
```

32. Write a program to extract the domain name , given an email id

Solution:

```
email = raw_input("Enter an email id")
username, domain = email.split("@")
print "Username is:", username
print "Domain is:", domain
```

33. Write a calculator program which takes in an expression and replies back with the right answer

Data:

```
input : 3 + 4
output : 7
```

Solution:

```
ex = raw_input("Enter an expression: ")
print eval(ex)
```

34. Write a program to read a csv file through csv module

Solution:

```
import csv
f = open("data.csv")
for row in csv.reader(f):
    print row
f.close()
```

35. Write a program to read a csv file without using csv module

Solution:

```
f = open("data.csv")
l = []
for line in f:
    l.append(line.split(","))

f.close()
print "Processed the csv file as:\n", l
```

36. Write a program to save a dictionary as json file

Solution:

```
import json
d = {"name":"Aditya", "email":"sp.aditya@gmail.com"}
s = json.dumps(d)
fh = open("data.json", 'wt')
fh.write(s)
fh.close()
```

37. Write a program to accept a string from the user and append it to a file

Solution:

```
fh = open("f1.txt", 'at')
s = raw_input("Enter any text that you want to be
appended to the file:")
fh.write(s)
fh.close()
```

38. Write a class called Account which has name, accountnumber, balance as attribute.

Solution:

```
class Account(object):
    def __init__(self, n, ac, b):
```

```
        self.name = n
        self.accnumber = ac
        self.balance = b
```

39. Write a program to create two Student objects and implement the equality operator such that two student objects compare as equals when studId and studName are one same

Solution:

```
class Student(object):
    def __init__(self, i, n):
        self.studId = i
        self.studName = n

class Student(object):
    def __init__(self, i, n):
        self.studId = i
        self.studName = n
    def __eq__(self, other):
        return self.studId == other.studId and
self.studName == other.studName

s1 = Student(1,'John Doe')
s2 = Student(1,'John Doe')
print s1 == s2
```

40. Write a program to write a list to a file

Solution:

```
l = ['one','two','three']
fh = open('listfile.txt', 'wt')
for line in l:
    fh.write(str(line) + "\n")
fh.close()
```

41. Write a program to find the size of a file

Solution:

```
import os
fn = raw_input("Enter a file name: ")
stat = os.stat("filename.txt")
print "Size of the file is ", stat.st_size
```

42. Given an error file, find the number of times error occurs in the file

Solution:

```
total = 0
fh = open('error.log', 'rt')
for line in fh:
    if "error" in line:
        total += 1
print "the word 'error' occurs ", total, "times"
```

43. Write program to accept two integers and return back with sum of all the integers between the two given integers (inclusive)

Solution:

```
s = input("Enter the beginning number")
e = input("Enter the last number")
print sum(range(s, e+1))
```

44. Write a python program to delete a given file name using check_call of subprocess module

```
import subprocess
subprocess.check_call('del a.txt', shell=True)
```

45. Write a python program to copy the contents of a file to another file

Solution:

```
import shutil
shutil.copyfile('file1.txt', 'file2.txt')
```

46. Write a python program to move a file to a different location

Solution:

```
import shutil
s = raw_input("Enter the source file name (you can
enter relative path or absolute path): ")
d = raw_input("Enter the destination file name (you can
enter relative path or absolute path): ")
shutil.move(s, d)
```

47. Write a python program to create a directory if it does not exist

Solution:

```
dirs = raw_input("Enter the directory path: ")
if not os.path.exists(dirs):
    os.makedirs(dirs)
```

48. Write a program to find the number of lines in a given file.

Solution:

```
fh = open('file.txt', 'rt')
count = 0
for line in fh:
    count += 1
fh.close()
print "There are a total of ", count, "lines in the
file"
```

49. Write a program to print all the leap years between two given years

Solution:

```
def isleap(n):
    leap = False
    if year %4 == 0:
        if year %100 ==0:
            if year % 400 == 0:
                leap = True
        else:
            lean = True
    return leap

s = input("Enter the first year greater than 1752:")
e = input("Enter the second year greater than 1752:")
l = range(s,e)
for y in l:
    if isleap(y):
        print y
```

50. Write a program to print all the prime number between two given integers

Solution:

```
def isprime(n):
    count = 2
    def isprime
    prime = True
    while count <= n / 2:
        if n % count == 0:
            prime = False
        if not prime:
            break
        count += 1
    return prime
```

```
s = input("Enter the first number:")
e = input("Enter the second number:")

l = range(s,e)
for n in l:
    if isprime(n):
        print n
```

51. Write a program to list all the numbers between 1 and hundred which are divisible by 2 and divisble by 13

Solution:

```
l = range(1,100)
for n in l:
    if n%2 ==0 and n%13 ==0:
        print n
```

52. Write a program to remove all the duplicates from the given list

Solution:

```
l = [1,2,3,4,4,5,1,6,8,2]
no_dupes = list(set(l))
print no_dupes
```

53. Loop through a dictionary and print key value pairs

Solution:

```
d = {'email':'sp.aditya@gmail', 'name':'Aditya'}

for k,v in d.items():
    print "Key", k, "has value", v
```

54. Easter egg to print out the zen of python

Solution:

```
import this
```

55. Write a program to provide the difference between the lowest and highest number in a list of numbers

Solution:

```
l = [1,2,4,5,6,7,8,9,10]

lowest = min(l)
highest = max(l)
print highest - lowest
```

56. Write a program to provide the centered average of a given list of integers. Centered average is the average of the numbers ignoring the smallest and the largest

Solution:

```
l = [1,2,4,5,6,7,8,9,10]

lowest = min(l)
highest = max(l)

count = 0
total = 0
for num in l:
    if num != lowest or num != highest:
        count += 1
        total += num
print "centered average : " , float(total)/count
```

57. Accept text from the user and print how many times each word occurs.

Solution:

```
s = raw_input("Enter some text")

l = s.split()

d = {}
for word in l:
    d[word] = l.count(word)
for k,v in d.items():
    print k,"occurs", v , "times"
```

58 . Take two string inputs and print the longer string

Solution:

```
s1 = raw_input("Enter your first string: ")
s2 = raw_input("Enter your second string: ")
if len(s1) == len(s2):
    print s1
    print s2
elif len(s1) > len(s2):
    print s1
else :
    print s2
```

59. Given the radius , write a program to print the area of a circle

Solution:

```
r = input("Enter the circle radius: ")
print "Area is ", float(22*r*r)/7
```

60. Generate 100 random numbers

Solution:

```
import random
l = []
```

```
for n in range(100):
    l.append(random.random())
```

61. Store the below contents in movie.json and print the poster url

Data:

```
{"Title":"Batman","Year":"1989","Rated":"PG-
13","Released":"23 Jun 1989","Runtime":"126
min","Genre":"Action, Adventure","Director":"Tim
Burton","Writer":"Bob Kane (Batman characters), Sam
Hamm (story), Sam Hamm (screenplay), Warren Skaaren
(screenplay)","Actors":"Michael Keaton, Jack Nicholson,
Kim Basinger, Robert Wuhl","Plot":"The Dark Knight of
Gotham City begins his war on crime with his first
major enemy being the clownishly homicidal
Joker.","Language":"English, French","Country":"USA,
UK","Awards":"Won 1 Oscar. Another 9 wins & 26
nominations.","Poster":"https://images-na.ssl-images-
amazon.com/images/M/MV5BMTYwNjAyODIyMF5BMl5BanBnXkFtZTY
wNDMwMDk2._V1_SX300.jpg","Ratings":[{"Source":"Internet
Movie Database","Value":"7.6/10"},{"Source":"Rotten
Tomatoes","Value":"72%"},{"Source":"Metacritic","Value"
:"69/100"}],"Metascore":"69","imdbRating":"7.6","imdbVo
tes":"291,045","imdbID":"tt0096895","Type":"movie","DVD
":"25 Mar 1997","BoxOffice":"N/A","Production":"Warner
Bros. Pictures","Website":"N/A","Response":"True"}
```

Solution:

```
import json
fh = open('movie.json')
s = fh.read()
data = json.loads(s)
print data['Poster']
```

62. Write a program to compress and decompress a string

Solution:

```
s = 'Hello how are you'
import zlib
zs = zlib.compress(s)

print "Compressed string", zs

us = zlib.uncompress(zs)
print "Uncompressed string", us
```

63. Write a program that taken in 3 numbers and prints the maximum of them

Solution:

```
n1 = input("Enter the first number: ")
n2 = input("Enter the second number: ")
n3 = input("Enter the third number: ")

print "The max of the three is ", max((n1,n2,n3))
```

64. Write a program that taken in 3 numbers and prints the minimum of them

Solution:

```
n1 = input("Enter the first number: ")
n2 = input("Enter the second number: ")
n3 = input("Enter the third number: ")

print "Minimum number of the three is ",
min((n1,n2,n3))
```

65. Write a program and print its pid

Solution:

```
import os
print os.getpid()
```

66. Write a function to create a unique file name

Solution:

```
import time
import os
def getfilename(ext):
    return time.time()+ os.getpid() + "."+ ext
```

67. Take two files and inform the user if the files are same or not.

Solution:

```
import filecmp
f1 = raw_input("Enter the first file name: ")
f2 = raw_input("Enter the second file name: ")

if filecmp.cmp(f1,f2):
    print "The files are same"
else:
    print "The files are different"
```

68. Write a program to print the current working directory

Solution:

```
import os
print "Current working directory is", os.getcwd()
```

69. Write a program to accept an environment variable and print its value if it exists

Solution:

```
import os
env_var = raw_input("Enter the environment variable: ")
env_value = os.getenv(env_var)
if env_value:
    print "The environment variable", env_var, "is
present and has the value:"
    print env_value
```

70. Write a program which accepts the file path and checks if the file is present

Solution:

```
import os.path
file_name= raw_input("Enter the file path")
if os.path.is_file(file_name):
    print "File exists"
else:
    print "The file is not present"
```

71. Write a program which accepts a path and checks if the path is a file or a directory

Solution:

```
import os.path

name = raw_input("Enter the path")
if os.path.isfile(name):
    print "Path exists and is a file"
elif os.path.isdir(name):
    print "Path exists and is a directory"
else:
    print "Path does not exist"
```

72. Write a program to print the ceil, floor, of an input number

Solution:

```
import math
n = input("Enter a number: ")

print math.ceil(n), math.floor(n)
```

73. Write a program to accept radius and print the circumference of a circle using the pi value present in math module

Solution:

```
import math
r = input("Enter the radius: ")
print math.pi * 2 * r
```

74. Write a program to print the current time in the below format

Data:
12/6/2017 0:11:36

Solution:

```
from datetime import datetime
now = datetime.now()
mm = str(now.month)
dd = str(now.day)
yyyy = str(now.year)
hour = str(now.hour)
mi = str(now.minute)
ss = str(now.second)

print mm + "/" + dd + "/" + yyyy + " " + hour + ":" +
mi + ":" + ss
```

75. Given three dictionaries write a program to merge them all into one dictionary

Solution:

```
d1 = {"name" : 'Aditya', 'email':'sp.aditya@gmail.com'}
d2 = {'name': 'Aditya S P', 'city': 'Bangalore'}
d3 = {'city':'Bengaluru','age': 35}

d4 = {}
d4.update(d2)
d4.update(d3)
d4.update(d1)

for k,v in d4.items():
    print "key ", k, 'has ', v
```

76. Give the two dictionaries, instead of merging the dictionaries, add the values of common keys

Solution:

```
d1 = {'John': 255, 'David': 300}
d2 = {'David': 325,'Jane':200}
d3 = {}
for k in d1:
    if k in d2:
        d3['k'] = d1[k] + d2[k]
    else
        d3[k] = d1[k]

for k,v in d2.items():
    if k not in d3:
        d3[k] = v
print d3
```

77. Write a program to print the index and the item of a list

Solution:

```
l = ['a','b','c','d']

for idx, item in enumerate(l):
    print item, 'in position', idx
```

78. In the given string, how can we insert the word 'of' in between the words discovery and India

Solution:

```
s = 'the discovery India'

l = s.split()
l.insert(2,'of')
print ' '.join(l)
```

79. Write a program to skip the first 100 bytes of a file and read further

Solution:

```
fn = raw_input("Enter the file name ")
fh = open(fn)
fh.seek(100)
s = fh.read()
print s
```

80. Given a list of temperatures in C convert them to F. COnversion function

Data:

```
f = C * 1.8 + 32
```

Solution:

```
l =[100,-32,100,45]

def ctof(x):
    return c * 1.8 + 32
```

```
l_in_f = map(ctof,l)
print l_in_f
```

81. Write a python program to construct the below pattern

Data:

```
**********
*********
********
*******
******
*****
****
***
**
*
```

Solution:

```
l = range(1,11)
l.reverse()
for i in l:
    s = ""
    for j in range(1,i):
        s += '*'
    print s
```

82. Write a program to check if the given program is already in sorted order

Solution:

```
l = [3,4,5,2,5]
l1 = [1,2,3,4,5]

s = sorted(l)
if s == l:
    print "list l is already in sorted order"
```

```
else:
    print "list l is not in sorted order"
s = sorted(l1)
if s == l1:
    print "list l1 is already in sorted order"
else:
    print "list l1 is not in sorted order"
```

83. Given a list of integers write a program to create a list which contains only the even numbers

Solution:

```
def iseven(x):
    return x %2 == 0

l = [567,24,78,23,89,34,65,23,67]

e = filter(iseven, l)
```

84. To the Complex class create above implement the functionality of addition

Solution:

```
    def __add__(self, other):
        return Complex(self.r + other.r , self.i +
other.i)
```

85. Write a program to print all the files in the directories and subdirectories from a given path

Solution:

```
p = raw_input("Enter a path: ")
import os

for dirName, subdirList, fileList in os.walk(p):
```

```
    print('Found directory: %s' % dirName)
    for fname in fileList:
        print('\t%s' % fname)
```

86. Create a list which contains only even numbers from 10 to 21

Solution:

```
l = range(10,21,2)
```

87. Create program to download the image at this particular url

Data:

```
https://images-na.ssl-images-
amazon.com/images/M/MV5BMTYwNjAyODIyMF5BMl5BanBnXkFtZTY
wNDMwMDk2._V1_SX300.jpg
```

Solution:

```
import requests
r = requests.get("https://images-na.ssl-images-
amazon.com/images/M/MV5BMTYwNjAyODIyMF5BMl5BanBnXkFtZTY
wNDMwMDk2._V1_SX300.jpg")
fh = open('image.jpg','wb')
fh.write(r.content)
fh.close()
```

88. Create a program to convert the below list to a list which has capitalized names

Data:

```
names = ['aditya','arun','john','shiv']
```

Solution:

```
def cap(x):
    return x.capitalize()

cap_names = map(cap, names)
```

```
for name in cap_names:
    print name
```

89. Write a program to generate a random integer between 1 and 100

Solution:

```
import random as r
def create():
    return int(r.uniform(1,100))

print create()
```

90. Write a program to generate a random integer between 1 and 100. Ask the user to input their guess. The program should then say if the guess is low or guess is high and continue to accept input untill the user gets it right

Solution:

```
import random as r
def create():
    return int(r.uniform(1,100))

r = create()
i = 0
guesses =0
print "I have generated a random integer between 1 and
100 inclusive. Guess the number."
print "Your guesses start now"
while True:
    guesses += 1
    i = input("Enter your guess:")
    if i < r:
        print "too low"
    elif i> r:
        print "too high"
    else :
        print "Good Guess. you took ", guesses,
"guesses!"
        break;
```

91. Given a string decode it using the below logic - Every character is replaced by a character two from its position

Data:

```
s = 'K"mpqy"{qw"iqv"kv#'
```

Solution:

```
s = 'K"mpqy"{qw"iqv"kv#'
def  leaptwo(x):
    return chr(ord(x)-2)
data = map(leaptwo, s)
print "".join(data)
```

92. Write a program to accept a few number as input from the user and calculate the average of the same

Solution:

```
l = []
while True:
    n = input("Enter a number (Enter 0 to terminate):
")
    if n == 0:
        break;
    l.append(n)

print "the average of the numbers entered is",
float(sum(l))/len(l)
```

93. Write a program to roll the dice! The program should print a number 1 to 6 everytime the user asks it to

Solution:

```
import random as r

while True:
    i = raw_input("Do you want to roll the dice[y/n]?")
    if i == 'n':
        break
    if i == 'y':
        n = r.randint(1,6)
```

```
        print n;
```

94. Write a program to check if a given number is even or odd

Solution:

```
def iseven(x):
    return x % 2 ==0

x = input("Enter a number: ")
if iseven(x):
    print "Its a even number"
else:
    print "Its an odd number"
```

95. Given the below text extract the values Fri, Oct and 28 using regular expressions

Data:

```
s = "Fri Oct 28 05:43:38.294803 2016"
```

Solution:

```
s = "Fri Oct 28 05:43:38.294803 2016"
p = r'^(?P<day>[a-zA-z]{3}) (?P<month>[a-zA-z]{3})
(?P<date>[\d]{1,2}) (?P<timestamp>\b[\d:\.]+\b)
(?P<year>[\d]{4})$'
import re
captured = re.findall(p,s)
print captured[0][0], captured[0][1], captured[0][2]
```

96. Write a program to accept input and calculate BMI. Accept metric values.

Data:

```
formula for BMI calculation is weight / (height *
height). Weight in kgs and height in meters
```

Solution:

```
weight = input("Enter your weight in kgs")
height = input("Enter your height in meters")
BMI = weight / (height ** 2)
```

```
print "Your BMI is ", BMI
```

97. Write a program to recursively calculate the sum to n numbers

Solution:

```
def sum(x):
    if x == 0 :
        return 0
    else:
        return x + sum(x-1)

print sum(5)
```

98. Write a non-recursive program to calculate the product to n numbers

Solution:

```
n = input("Enter an integer: ")
prod = 1
for x in range(1,n+1):
    prod = prod * x
print prod
```

99. Write a program which print 5 numbers. But the program should wait atleast 5 seconds before it prints the next number

Solution:

```
    import time
    for i in range(5)
        print i
        time.sleep(5)
```

100. Start with 1 and increment the number by 1 for 10000000 times. Time how much it takes to perform this operation

Solution:

```
import time
```

```
def toinc()
    x = 1
    count = 0
    while count < 10000000:
        count +=1
    print "Incremented to ", count

s = time.time()
toinc()
e = time.time()
print "It took", e -s
```

101. Write a program to find out the left most digit of a number:

Solution:

```
n = int(input("Enter a huge number"))
while n > 10:
    n = n/10
print n
```