100+ Python challenging programming exercises

1. Level description
Level Description
Level 1 Beginner means someone who has just gone through an introductory Python course. He can solve some problems with 1 or 2 Python classes or functions. Normally, the answers could directly be found in the textbooks.
Level 2 Intermediate means someone who has just learned Python, but already has a relatively strong programming background from before. He should be able to solve problems which may involve 3 or 3 Python classes or functions. The answers cannot be directly be found in the textbooks.
Level 3 Advanced. He should use Python to solve more complex problem using more rich libraries functions and data structures and algorithms. He is supposed to solve the problem using several Python standard packages and advanced techniques.

2. Problem template

#--#
Question
Hints
Solution

3. Questions

#--#
Question 1
Level 1

Question:
Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).
The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints:
Consider use range(#begin, #end) method

Solution:
```
l=[]
for i in range(2000, 3201):
    if (i%7==0) and (i%5!=0):
        l.append(str(i))

print ','.join(l)
```
#--#

#--#
Question 2
Level 1

Question:
Write a program which can compute the factorial of a given numbers.
The results should be printed in a comma-separated sequence on a single line.
Suppose the following input is supplied to the program:
8

Then, the output should be:
40320

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
def fact(x):
    if x == 0:
        return 1
    return x * fact(x - 1)

x=int(raw_input())
print fact(x)
```
#--#

#--#
Question 3
Level 1

Question:
With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary.
Suppose the following input is supplied to the program:
8
Then, the output should be:
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.
Consider use dict()

Solution:
```
n=int(raw_input())
d=dict()
for i in range(1,n+1):
    d[i]=i*i

print d
```
#--#

#--#
Question 4
Level 1

Question:
Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.
Suppose the following input is supplied to the program:
34,67,55,33,12,98
Then, the output should be:
['34', '67', '55', '33', '12', '98']

('34', '67', '55', '33', '12', '98')

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.
tuple() method can convert list to tuple

Solution:
```
values=raw_input()
l=values.split(",")
t=tuple(l)
print l
print t
#---------------------------------------#
```

```
#---------------------------------------#
```
Question 5
Level 1

Question:
Define a class which has at least two methods:
getString: to get a string from console input
printString: to print the string in upper case.
Also please include simple test function to test the class methods.

Hints:
Use __init__ method to construct some parameters

Solution:
```
class InputOutString(object):
    def __init__(self):
        self.s = ""

    def getString(self):
        self.s = raw_input()

    def printString(self):
        print self.s.upper()

strObj = InputOutString()
strObj.getString()
strObj.printString()
#---------------------------------------#
```

```
#---------------------------------------#
```
Question 6
Level 2

Question:
Write a program that calculates and prints the value according to the given formula:
Q = Square root of [(2 * C * D)/H]
Following are the fixed values of C and H:
C is 50. H is 30.

D is the variable whose values should be input to your program in a comma-separated sequence.
Example
Let us assume the following comma separated input sequence is given to the program:
100,150,180
The output of the program should be:
18,22,24

Hints:
If the output received is in decimal form, it should be rounded off to its nearest value (for example, if the output received is 26.0, it should be printed as 26)
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
#!/usr/bin/env python
import math
c=50
h=30
value = []
items=[x for x in raw_input().split(',')]
for d in items:
    value.append(str(int(round(math.sqrt(2*c*float(d)/h)))))

print ','.join(value)
#---------------------------------------#

#---------------------------------------#
```

Question 7
Level 2

Question:
Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.
Note: i=0,1.., X-1; j=0,1,¡
Example
Suppose the following inputs are given to the program:
3,5
Then, the output of the program should be:
[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

Hints:
Note: In case of input data being supplied to the question, it should be assumed to be a console input in a comma-separated form.

Solution:

```
input_str = raw_input()
dimensions=[int(x) for x in input_str.split(',')]
rowNum=dimensions[0]
colNum=dimensions[1]
multilist = [[0 for col in range(colNum)] for row in range(rowNum)]

for row in range(rowNum):
    for col in range(colNum):
```

```
        multilist[row][col]= row*col

print multilist
#-------------------------------------#
```

```
#-------------------------------------#
```
Question 8
Level 2

Question:
Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.
Suppose the following input is supplied to the program:
without,hello,bag,world
Then, the output should be:
bag,hello,without,world

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
items=[x for x in raw_input().split(',')]
items.sort()
print ','.join(items)
#-------------------------------------#
```

```
#-------------------------------------#
```
Question 9
Level 2

Question£º
Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.
Suppose the following input is supplied to the program:
Hello world
Practice makes perfect
Then, the output should be:
HELLO WORLD
PRACTICE MAKES PERFECT

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
lines = []
while True:
    s = raw_input()
    if s:
        lines.append(s.upper())
    else:
        break;
```

```
for sentence in lines:
    print sentence
#------------------------------------#
```

```
#------------------------------------#
```

Question 10
Level 2

Question:
Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.
Suppose the following input is supplied to the program:
hello world and practice makes perfect and hello world again
Then, the output should be:
again and hello makes perfect practice world

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.
We use set container to remove duplicated data automatically and then use sorted() to sort the data.

Solution:

```
s = raw_input()
words = [word for word in s.split(" ")]
print " ".join(sorted(list(set(words))))
#------------------------------------#
```

```
#------------------------------------#
```

Question 11
Level 2

Question:
Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence.
Example:
0100,0011,1010,1001
Then the output should be:
1010
Notes: Assume the data is input by console.

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
value = []
items=[x for x in raw_input().split(',')]
for p in items:
    intp = int(p, 2)
    if not intp%5:
        value.append(p)

print ','.join(value)
```

#--#

#--#
Question 12
Level 2

Question:
Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number.
The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
values = []
for i in range(1000, 3001):
    s = str(i)
    if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])%2==0):
        values.append(s)
print ",".join(values)
```
#--#

#--#
Question 13
Level 2

Question:
Write a program that accepts a sentence and calculate the number of letters and digits.
Suppose the following input is supplied to the program:
hello world! 123
Then, the output should be:
LETTERS 10
DIGITS 3

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
s = raw_input()
d={"DIGITS":0, "LETTERS":0}
for c in s:
    if c.isdigit():
        d["DIGITS"]+=1
    elif c.isalpha():
        d["LETTERS"]+=1
    else:
        pass
print "LETTERS", d["LETTERS"]
print "DIGITS", d["DIGITS"]
```
#--#

#--#
Question 14
Level 2

Question:
Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters.
Suppose the following input is supplied to the program:
Hello world!
Then, the output should be:
UPPER CASE 1
LOWER CASE 9

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
s = raw_input()
d={"UPPER CASE":0, "LOWER CASE":0}
for c in s:
    if c.isupper():
        d["UPPER CASE"]+=1
    elif c.islower():
        d["LOWER CASE"]+=1
    else:
        pass
print "UPPER CASE", d["UPPER CASE"]
print "LOWER CASE", d["LOWER CASE"]
```
#--#

#--#
Question 15
Level 2

Question:
Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.
Suppose the following input is supplied to the program:
9
Then, the output should be:
11106

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
a = raw_input()
n1 = int( "%s" % a )
n2 = int( "%s%s" % (a,a) )
n3 = int( "%s%s%s" % (a,a,a) )
n4 = int( "%s%s%s%s" % (a,a,a,a) )
print n1+n2+n3+n4
```
#--#

#--#
Question 16
Level 2

Question:
Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers.
Suppose the following input is supplied to the program:
1,2,3,4,5,6,7,8,9
Then, the output should be:
1,3,5,7,9

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
values = raw_input()
numbers = [x for x in values.split(",") if int(x)%2!=0]
print ",".join(numbers)
```
#--#

Question 17
Level 2

Question:
Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:
D 100
W 200
¡□
D means deposit while W means withdrawal.
Suppose the following input is supplied to the program:
D 300
D 300
W 200
D 100
Then, the output should be:
500

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:
```
import sys
netAmount = 0
while True:
    s = raw_input()
    if not s:
        break
    values = s.split(" ")
    operation = values[0]
    amount = int(values[1])
```

```
    if operation=="D":
        netAmount+=amount
    elif operation=="W":
        netAmount-=amount
    else:
        pass
print netAmount
#---------------------------------------#
```

```
#---------------------------------------#
```

Question 18
Level 3

Question:
A website requires the users to input username and password to register. Write a program to check the validity of password input by users.
Following are the criteria for checking the password:
1. At least 1 letter between [a-z]
2. At least 1 number between [0-9]
1. At least 1 letter between [A-Z]
3. At least 1 character from [$#@]
4. Minimum length of transaction password: 6
5. Maximum length of transaction password: 12
Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.
Example
If the following passwords are given as input to the program:
ABd1234@1,a F1#,2w3E*,2We3345
Then, the output of the program should be:
ABd1234@1

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solutions:

```
import re
value = []
items=[x for x in raw_input().split(',')]
for p in items:
    if len(p)<6 or len(p)>12:
        continue
    else:
        pass
    if not re.search("[a-z]",p):
        continue
    elif not re.search("[0-9]",p):
        continue
    elif not re.search("[A-Z]",p):
        continue
    elif not re.search("[$#@]",p):
        continue
    elif re.search("\s",p):
```

```
        continue
    else:
        pass
    value.append(p)
print ",".join(value)
```
#--#

#--#
Question 19
Level 3

Question:
You are required to write a program to sort the (name, age, height) tuples by ascending order where name is string, age and height are numbers. The tuples are input by console. The sort criteria is:
1: Sort based on name;
2: Then sort based on age;
3: Then sort by score.
The priority is that name > age > score.
If the following tuples are given as input to the program:
Tom,19,80
John,20,90
Jony,17,91
Jony,17,93
Json,21,85
Then, the output of the program should be:
[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.
We use itemgetter to enable multiple sort keys.

Solutions:
```
from operator import itemgetter, attrgetter

l = []
while True:
    s = raw_input()
    if not s:
        break
    l.append(tuple(s.split(",")))

print sorted(l, key=itemgetter(0,1,2))
```
#--#

#--#
Question 20
Level 3

Question:
Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

Hints:
Consider use yield

Solution:

```
def putNumbers(n):
    i = 0
    while i<n:
        j=i
        i=i+1
        if j%7==0:
            yield j

for i in reverse(100):
    print i
```

#----------#

#----------#

Question 21
Level 3

Question£º
A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with a given steps. The trace of robot movement is shown as the following:
UP 5
DOWN 3
LEFT 3
RIGHT 2
¡□
The numbers after the direction are steps. Please write a program to compute the distance from current position after a sequence of movement and original point. If the distance is a float, then just print the nearest integer.
Example:
If the following tuples are given as input to the program:
UP 5
DOWN 3
LEFT 3
RIGHT 2
Then, the output of the program should be:
2

Hints:
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
import math
pos = [0,0]
while True:
    s = raw_input()
    if not s:
        break
    movement = s.split(" ")
    direction = movement[0]
    steps = int(movement[1])
```

```
    if direction=="UP":
        pos[0]+=steps
    elif direction=="DOWN":
        pos[0]-=steps
    elif direction=="LEFT":
        pos[1]-=steps
    elif direction=="RIGHT":
        pos[1]+=steps
    else:
        pass

print int(round(math.sqrt(pos[1]**2+pos[0]**2)))
#---------------------------------------#
```

```
#---------------------------------------#
```

Question 22
Level 3

Question:
Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.
Suppose the following input is supplied to the program:
New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.
Then, the output should be:
2:2
3.:1
3?:1
New:1
Python:5
Read:1
and:1
between:1
choosing:1
or:2
to:1

Hints
In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
freq = {}   # frequency of words in text
line = raw_input()
for word in line.split():
    freq[word] = freq.get(word,0)+1

words = freq.keys()
words.sort()

for w in words:
    print "%s:%d" % (w,freq[w])
#---------------------------------------#
```

#--#
Question 23
level 1

Question:
    Write a method which can calculate square value of number

Hints:
    Using the ** operator

Solution:
```
def square(num):
    return num ** 2

print square(2)
print square(3)
```
#--#

#--#
Question 24
Level 1

Question:
    Python has many built-in functions, and if you do not know how to use it, you can read document online or find some books. But Python has a built-in document function for every built-in functions.
    Please write a program to print some Python built-in functions documents, such as abs(), int(), raw_input()
    And add document for your own function

Hints:
    The built-in document method is __doc__

Solution:
```
print abs.__doc__
print int.__doc__
print raw_input.__doc__

def square(num):
    '''Return the square value of the input number.

    The input number must be integer.
    '''
    return num ** 2

print square(2)
print square.__doc__
```
#--#

#--#
Question 25
Level 1

Question:

Define a class, which have a class parameter and have a same instance parameter.

Hints:
Define a instance parameter, need add it in __init__ method
You can init a object with construct parameter or set the value later

Solution:

```
class Person:
    # Define the class parameter "name"
    name = "Person"

    def __init__(self, name = None):
        # self.name is the instance parameter
        self.name = name

jeffrey = Person("Jeffrey")
print "%s name is %s" % (Person.name, jeffrey.name)

nico = Person()
nico.name = "Nico"
print "%s name is %s" % (Person.name, nico.name)
```

#----------#

#----------#

Question:
Define a function which can compute the sum of two numbers.

Hints:
Define a function with two numbers as arguments. You can compute the sum in the function and return the value.

Solution

```
def SumFunction(number1, number2):
 return number1+number2

print SumFunction(1,2)
```

#----------#

Question:
Define a function that can convert a integer into a string and print it in console.

Hints:

Use str() to convert a number to string.

Solution

```
def printValue(n):
 print str(n)

printValue(3)
```

#----------#

Question:
Define a function that can convert a integer into a string and print it in console.

Hints:

Use str() to convert a number to string.

Solution
```
def printValue(n):
 print str(n)

printValue(3)
```

#----------#
2.10

Question:
Define a function that can receive two integral numbers in string form and compute their sum and then print it in console.

Hints:

Use int() to convert a string to integer.

Solution
```
def printValue(s1,s2):
 print int(s1)+int(s2)

printValue("3","4") #7
```

#----------#
2.10

Question:
Define a function that can accept two strings as input and concatenate them and then print it in console.

Hints:

Use + to concatenate the strings

Solution
```
def printValue(s1,s2):
 print s1+s2

printValue("3","4") #34
```

#----------#
2.10

Question:
Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print al l strings line by line.

Hints:

Use len() function to get the length of a string

Solution

```
def printValue(s1,s2):
 len1 = len(s1)
 len2 = len(s2)
 if len1>len2:
  print s1
 elif len2>len1:
  print s2
 else:
  print s1
  print s2


printValue("one","three")
```

#----------#

2.10

Question:
Define a function that can accept an integer number as input and print the "It is an even number" if the number is even, otherwise print "It is an odd number".

Hints:

Use % operator to check if a number is even or odd.

Solution

```
def checkValue(n):
 if n%2 == 0:
  print "It is an even number"
 else:
  print "It is an odd number"


checkValue(7)
```

#----------#

2.10

Question:
Define a function which can print a dictionary where the keys are numbers between 1 and 3 (both included) and

the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary.
Use ** operator to get power of a number.

Solution

```
def printDict():
 d=dict()
 d[1]=1
 d[2]=2**2
 d[3]=3**2
 print d


printDict()
```

#----------#

2.10

Question:
Define a function which can print a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary.
Use ** operator to get power of a number.
Use range() for loops.

Solution

```
def printDict():
 d=dict()
 for i in range(1,21):
  d[i]=i**2
 print d


printDict()
```

#----------#

2.10

Question:
Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys. The function should just print the values only.

Hints:

Use dict[key]=value pattern to put entry into a dictionary.
Use ** operator to get power of a number.
Use range() for loops.
Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.

Solution

```
def printDict():
 d=dict()
 for i in range(1,21):
  d[i]=i**2
 for (k,v) in d.items():
  print v


printDict()
```

#--#
2.10

Question:
Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys. The function should just print the keys only.

Hints:

Use dict[key]=value pattern to put entry into a dictionary.
Use ** operator to get power of a number.
Use range() for loops.
Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.

Solution

```
def printDict():
 d=dict()
 for i in range(1,21):
  d[i]=i**2
 for k in d.keys():
  print k


printDict()
```

#--#
2.10

Question:
Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (both included).

Hints:

Use ** operator to get power of a number.
Use range() for loops.
Use list.append() to add values into a list.

Solution
```
def printList():
 li=list()
 for i in range(1,21):
  li.append(i**2)
 print li


printList()
```

#--#
2.10

Question:
Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function needs to print the first 5 elements in the list.

Hints:

Use ** operator to get power of a number.
Use range() for loops.
Use list.append() to add values into a list.
Use [n1:n2] to slice a list

Solution
```
def printList():
 li=list()
 for i in range(1,21):
  li.append(i**2)
 print li[:5]


printList()
```

#--#
2.10

Question:
Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function needs to print the last 5 elements in the list.

Hints:

Use ** operator to get power of a number.
Use range() for loops.

Use list.append() to add values into a list.
Use [n1:n2] to slice a list

Solution
```
def printList():
 li=list()
 for i in range(1,21):
  li.append(i**2)
 print li[-5:]


printList()
```

#--#
2.10

Question:
Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function needs to print all values except the first 5 elements in the list.

Hints:

Use ** operator to get power of a number.
Use range() for loops.
Use list.append() to add values into a list.
Use [n1:n2] to slice a list

Solution
```
def printList():
 li=list()
 for i in range(1,21):
  li.append(i**2)
 print li[5:]


printList()
```

#--#
2.10

Question:
Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20 (both included).

Hints:

Use ** operator to get power of a number.
Use range() for loops.
Use list.append() to add values into a list.
Use tuple() to get a tuple from a list.

Solution

```
def printTuple():
 li=list()
 for i in range(1,21):
  li.append(i**2)
 print tuple(li)

printTuple()
```

#----------#

2.10

Question:
With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.

Hints:

Use [n1:n2] notation to get a slice from a tuple.

Solution

```
tp=(1,2,3,4,5,6,7,8,9,10)
tp1=tp[:5]
tp2=tp[5:]
print tp1
print tp2
```

#----------#

2.10

Question:
Write a program to generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

Hints:

Use "for" to iterate the tuple
Use tuple() to generate a tuple from a list.

Solution

```
tp=(1,2,3,4,5,6,7,8,9,10)
li=list()
for i in tp:
 if tp[i]%2==0:
  li.append(tp[i])

tp2=tuple(li)
print tp2
```

#--#
2.14

Question:
Write a program which accepts a string as input to print "Yes" if the string is "yes" or "YES" or "Yes", otherwise print "No".

Hints:

Use if statement to judge condition.

Solution
```
s= raw_input()
if s=="yes" or s=="YES" or s=="Yes":
   print "Yes"
else:
   print "No"
```

#--#
3.4

Question:
Write a program which can filter even numbers in a list by using filter function. The list is: [1,2,3,4,5,6,7,8,9,10].

Hints:

Use filter() to filter some elements in a list.
Use lambda to define anonymous functions.

Solution
```
li = [1,2,3,4,5,6,7,8,9,10]
evenNumbers = filter(lambda x: x%2==0, li)
print evenNumbers
```

#--#
3.4

Question:
Write a program which can map() to make a list whose elements are square of elements in [1,2,3,4,5,6,7,8,9,10].

Hints:

Use map() to generate a list.
Use lambda to define anonymous functions.

Solution

```
li = [1,2,3,4,5,6,7,8,9,10]
squaredNumbers = map(lambda x: x**2, li)
print squaredNumbers
```

#----------#

3.5

Question:
Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10].

Hints:

Use map() to generate a list.
Use filter() to filter elements of a list.
Use lambda to define anonymous functions.

Solution

```
li = [1,2,3,4,5,6,7,8,9,10]
evenNumbers = map(lambda x: x**2, filter(lambda x: x%2==0, li))
print evenNumbers
```

#----------#

3.5

Question:
Write a program which can filter() to make a list whose elements are even number between 1 and 20 (both included).

Hints:

Use filter() to filter elements of a list.
Use lambda to define anonymous functions.

Solution

```
evenNumbers = filter(lambda x: x%2==0, range(1,21))
print evenNumbers
```

#----------#

3.5

Question:
Write a program which can map() to make a list whose elements are square of numbers between 1 and 20 (both included).

Hints:

Use map() to generate a list.

Use lambda to define anonymous functions.

Solution
```
squaredNumbers = map(lambda x: x**2, range(1,21))
print squaredNumbers
```

#--#
7.2

Question:
Define a class named American which has a static method called printNationality.

Hints:

Use @staticmethod decorator to define class static method.

Solution
```
class American(object):
    @staticmethod
    def printNationality():
        print "America"

anAmerican = American()
anAmerican.printNationality()
American.printNationality()
```

#--#

7.2

Question:
Define a class named American and its subclass NewYorker.

Hints:

Use class Subclass(ParentClass) to define a subclass.

Solution:

```
class American(object):
    pass

class NewYorker(American):
    pass

anAmerican = American()
```

```
aNewYorker = NewYorker()
print anAmerican
print aNewYorker
```

#--#

7.2

Question:
Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.

Hints:

Use def methodName(self) to define a method.

Solution:

```
class Circle(object):
    def __init__(self, r):
        self.radius = r

    def area(self):
        return self.radius**2*3.14

aCircle = Circle(2)
print aCircle.area()
```

#--#

7.2

Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.

Hints:

Use def methodName(self) to define a method.

Solution:

```
class Rectangle(object):
    def __init__(self, l, w):
```

```
        self.length = l
        self.width  = w

    def area(self):
        return self.length*self.width

aRectangle = Rectangle(2,10)
print aRectangle.area()
```

#--#

7.2

Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default.

Hints:

To override a method in super class, we can define a method with the same name in the super class.

Solution:

```
class Shape(object):
    def __init__(self):
        pass

    def area(self):
        return 0

class Square(Shape):
    def __init__(self, l):
        Shape.__init__(self)
        self.length = l

    def area(self):
        return self.length*self.length

aSquare= Square(3)
print aSquare.area()
```

#--#

Please raise a RuntimeError exception.

Hints:

Use raise() to raise an exception.

Solution:

raise RuntimeError('something wrong')

#--#
Write a function to compute 5/0 and use try/except to catch the exceptions.

Hints:

Use try/except to catch exceptions.

Solution:

```
def throws():
    return 5/0

try:
    throws()
except ZeroDivisionError:
    print "division by zero!"
except Exception, err:
    print 'Caught an exception'
finally:
    print 'In finally block for cleanup'
```

#--#
Define a custom exception class which takes a string message as attribute.

Hints:

To define a custom exception, we need to define a class inherited from Exception.

Solution:

```
class MyError(Exception):
    """My own exception class

    Attributes:
        msg  -- explanation of the error
    """
```

```
    def __init__(self, msg):
        self.msg = msg

error = MyError("something wrong")
```

#----------#
Question:

Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

Example:
If the following email address is given as input to the program:

john@google.com

Then, the output of the program should be:

john

In case of input data being supplied to the question, it should be assumed to be a console input.

Hints:

Use \w to match letters.

Solution:

```
import re
emailAddress = raw_input()
pat2 = "(\w+)@((\w+\.)+(com))"
r2 = re.match(pat2,emailAddress)
print r2.group(1)
```

#----------#
Question:

Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the company name of a given email address. Both user names and company names are composed of letters only.

Example:
If the following email address is given as input to the program:

john@google.com

Then, the output of the program should be:

google

In case of input data being supplied to the question, it should be assumed to be a console input.

Hints:

Use \w to match letters.

Solution:
```
import re
emailAddress = raw_input()
pat2 = "(\w+)@(\w+)\.(com)"
r2 = re.match(pat2,emailAddress)
print r2.group(2)
```

#--#
Question:

Write a program which accepts a sequence of words separated by whitespace as input to print the words composed of digits only.

Example:
If the following words is given as input to the program:

2 cats and 3 dogs.

Then, the output of the program should be:

['2', '3']

In case of input data being supplied to the question, it should be assumed to be a console input.

Hints:

Use re.findall() to find all substring using regex.

Solution:
```
import re
s = raw_input()
print re.findall("\d+",s)
```

#--#
Question:

Print a unicode string "hello world".

Hints:

Use u'strings' format to define unicode string.

Solution:

```
unicodeString = u"hello world!"
print unicodeString
```

#--#
Write a program to read an ASCII string and to convert it to a unicode string encoded by utf-8.

Hints:

Use unicode() function to convert.

Solution:

```
s = raw_input()
u = unicode( s ,"utf-8")
print u
```

#--#
Question:

Write a special comment to indicate a Python source code file is in unicode.

Hints:

Solution:

```
# -*- coding: utf-8 -*-
```

#--#
Question:

Write a program to compute 1/2+2/3+3/4+...+n/n+1 with a given n input by console (n>0).

Example:
If the following n is given as input to the program:

5

Then, the output of the program should be:

3.55

In case of input data being supplied to the question, it should be assumed to be a console input.

Hints:
Use float() to convert an integer to a float

Solution:

```
n=int(raw_input())
sum=0.0
```

```
for i in range(1,n+1):
    sum += float(float(i)/(i+1))
print sum
```

#--#

Question:

Write a program to compute:

f(n)=f(n-1)+100 when n>0
and f(0)=1

with a given n input by console (n>0).

Example:
If the following n is given as input to the program:

5

Then, the output of the program should be:

500

In case of input data being supplied to the question, it should be assumed to be a console input.

Hints:
We can define recursive function in Python.

Solution:

```
def f(n):
    if n==0:
        return 0
    else:
        return f(n-1)+100

n=int(raw_input())
print f(n)
```

#--#

Question:

The Fibonacci Sequence is computed based on the following formula:

f(n)=0 if n=0
f(n)=1 if n=1
f(n)=f(n-1)+f(n-2) if n>1

Please write a program to compute the value of f(n) with a given n input by console.

Example:
If the following n is given as input to the program:

7

Then, the output of the program should be:

13

In case of input data being supplied to the question, it should be assumed to be a console input.

Hints:
We can define recursive function in Python.

Solution:

```
def f(n):
    if n == 0: return 0
    elif n == 1: return 1
    else: return f(n-1)+f(n-2)

n=int(raw_input())
print f(n)
```

#--#

#--#

Question:

The Fibonacci Sequence is computed based on the following formula:

f(n)=0 if n=0
f(n)=1 if n=1
f(n)=f(n-1)+f(n-2) if n>1

Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given n input by console.

Example:
If the following n is given as input to the program:

7

Then, the output of the program should be:

0,1,1,2,3,5,8,13

Hints:
We can define recursive function in Python.
Use list comprehension to generate a list from an existing list.
Use string.join() to join a list of strings.

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
def f(n):
    if n == 0: return 0
    elif n == 1: return 1
    else: return f(n-1)+f(n-2)

n=int(raw_input())
values = [str(f(x)) for x in range(0, n+1)]
print ",".join(values)
```

#--#

Question:

Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.

Example:
If the following n is given as input to the program:

10

Then, the output of the program should be:

0,2,4,6,8,10

Hints:
Use yield to produce the next value in generator.

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
def EvenGenerator(n):
    i=0
    while i<=n:
        if i%2==0:
            yield i
        i+=1
```

```
n=int(raw_input())
values = []
for i in EvenGenerator(n):
    values.append(str(i))

print ",".join(values)
```

#--#

Question:

Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while n is input by console.

Example:
If the following n is given as input to the program:

100

Then, the output of the program should be:

0,35,70

Hints:
Use yield to produce the next value in generator.

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
def NumGenerator(n):
    for i in range(n+1):
        if i%5==0 and i%7==0:
            yield i

n=int(raw_input())
values = []
for i in NumGenerator(n):
    values.append(str(i))

print ",".join(values)
```

#--#

Question:

Please write assert statements to verify that every number in the list [2,4,6,8] is even.

Hints:
Use "assert expression" to make assertion.

Solution:

```
li = [2,4,6,8]
for i in li:
    assert i%2==0
```

#--#
Question:

Please write a program which accepts basic mathematic expression from console and print the evaluation result.

Example:
If the following string is given as input to the program:

35+3

Then, the output of the program should be:

38

Hints:
Use eval() to evaluate an expression.

Solution:

```
expression = raw_input()
print eval(expression)
```

#--#
Question:

Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

Hints:
Use if/elif to deal with conditions.

Solution:

```
import math
def bin_search(li, element):
    bottom = 0
```

```
    top = len(li)-1
    index = -1
    while top>=bottom and index==-1:
        mid = int(math.floor((top+bottom)/2.0))
        if li[mid]==element:
            index = mid
        elif li[mid]>element:
            top = mid-1
        else:
            bottom = mid+1

    return index

li=[2,5,7,9,11,17,222]
print bin_search(li,11)
print bin_search(li,12)
```

#--#

Question:

Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

Hints:

Use if/elif to deal with conditions.

Solution:

```
import math
def bin_search(li, element):
    bottom = 0
    top = len(li)-1
    index = -1
    while top>=bottom and index==-1:
        mid = int(math.floor((top+bottom)/2.0))
        if li[mid]==element:
            index = mid
        elif li[mid]>element:
            top = mid-1
        else:
            bottom = mid+1

    return index

li=[2,5,7,9,11,17,222]
print bin_search(li,11)
print bin_search(li,12)
```

#--#
Question:

Please generate a random float where the value is between 10 and 100 using Python math module.

Hints:
Use random.random() to generate a random float in [0,1].

Solution:

```
import random
print random.random()*100
```

#--#
Question:

Please generate a random float where the value is between 5 and 95 using Python math module.

Hints:
Use random.random() to generate a random float in [0,1].

Solution:

```
import random
print random.random()*100-5
```

#--#
Question:

Please write a program to output a random even number between 0 and 10 inclusive using random module and list comprehension.

Hints:
Use random.choice() to a random element from a list.

Solution:

```
import random
```

```
print random.choice([i for i in range(11) if i%2==0])
```

#--#
Question:

Please write a program to output a random number, which is divisible by 5 and 7, between 0 and 10 inclusive using random module and list comprehension.

Hints:
Use random.choice() to a random element from a list.

Solution:

```
import random
print random.choice([i for i in range(201) if i%5==0 and i%7==0])
```

#--#

Question:

Please write a program to generate a list with 5 random numbers between 100 and 200 inclusive.

Hints:
Use random.sample() to generate a list of random values.

Solution:

```
import random
print random.sample(range(100), 5)
```

#--#
Question:

Please write a program to randomly generate a list with 5 even numbers between 100 and 200 inclusive.

Hints:
Use random.sample() to generate a list of random values.

Solution:

```
import random
print random.sample([i for i in range(100,201) if i%2==0], 5)
```

#--#

Question:

Please write a program to randomly generate a list with 5 numbers, which are divisible by 5 and 7 , between 1 and 1000 inclusive.

Hints:
Use random.sample() to generate a list of random values.

Solution:

```
import random
print random.sample([i for i in range(1,1001) if i%5==0 and i%7==0], 5)
```

#--#

Question:

Please write a program to randomly print a integer number between 7 and 15 inclusive.

Hints:
Use random.randrange() to a random integer in a given range.

Solution:

```
import random
print random.randrange(7,16)
```

#--#

Question:

Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".

Hints:
Use zlib.compress() and zlib.decompress() to compress and decompress a string.

Solution:

```
import zlib
s = 'hello world!hello world!hello world!hello world!'
t = zlib.compress(s)
print t
print zlib.decompress(t)
```

#--#
Question:

Please write a program to print the running time of execution of "1+1" for 100 times.

Hints:
Use timeit() function to measure the running time.

Solution:

```
from timeit import Timer
t = Timer("for i in range(100):1+1")
print t.timeit()
```

#--#
Question:

Please write a program to shuffle and print the list [3,6,7,8].

Hints:
Use shuffle() function to shuffle a list.

Solution:

```
from random import shuffle
li = [3,6,7,8]
shuffle(li)
print li
```

#--#
Question:

Please write a program to shuffle and print the list [3,6,7,8].

Hints:
Use shuffle() function to shuffle a list.

Solution:

```
from random import shuffle
li = [3,6,7,8]
shuffle(li)
print li
```

#--#
Question:

Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey","Football"].

Hints:
Use list[index] notation to get a element from a list.

Solution:

```
subjects=["I", "You"]
verbs=["Play", "Love"]
objects=["Hockey","Football"]
for i in range(len(subjects)):
    for j in range(len(verbs)):
        for k in range(len(objects)):
            sentence = "%s %s %s." % (subjects[i], verbs[j], objects[k])
            print sentence
```

#--#
Please write a program to print the list after removing delete even numbers in [5,6,77,45,22,12,24].

Hints:
Use list comprehension to delete a bunch of element from a list.

Solution:

```
li = [5,6,77,45,22,12,24]
li = [x for x in li if x%2!=0]
print li
```

#--#
Question:

By using list comprehension, please write a program to print the list after removing delete numbers which are divisible by 5 and 7 in [12,24,35,70,88,120,155].

Hints:
Use list comprehension to delete a bunch of element from a list.

Solution:

```
li = [12,24,35,70,88,120,155]
```

```
li = [x for x in li if x%5!=0 and x%7!=0]
print li
```

#--#

Question:

By using list comprehension, please write a program to print the list after removing the 0th, 2nd, 4th,6th numbers in [12,24,35,70,88,120,155].

Hints:
Use list comprehension to delete a bunch of element from a list.
Use enumerate() to get (index, value) tuple.

Solution:

```
li = [12,24,35,70,88,120,155]
li = [x for (i,x) in enumerate(li) if i%2!=0]
print li
```

#--#

Question:

By using list comprehension, please write a program generate a 3*5*8 3D array whose each element is 0.

Hints:
Use list comprehension to make an array.

Solution:

```
array = [[ [0 for col in range(8)] for col in range(5)] for row in range(3)]
print array
```

#--#

Question:

By using list comprehension, please write a program to print the list after removing the 0th,4th,5th numbers in [12,24,35,70,88,120,155].

Hints:
Use list comprehension to delete a bunch of element from a list.
Use enumerate() to get (index, value) tuple.

Solution:

```
li = [12,24,35,70,88,120,155]
li = [x for (i,x) in enumerate(li) if i not in (0,4,5)]
print li
```

#--#

Question:

By using list comprehension, please write a program to print the list after removing the value 24 in [12,24,35,24,88,120,155].

Hints:
Use list's remove method to delete a value.

Solution:

```
li = [12,24,35,24,88,120,155]
li = [x for x in li if x!=24]
print li
```

#--#
Question:

With two given lists [1,3,6,78,35,55] and [12,24,35,24,88,120,155], write a program to make a list whose elements are intersection of the above given lists.

Hints:
Use set() and "&=" to do set intersection operation.

Solution:

```
set1=set([1,3,6,78,35,55])
set2=set([12,24,35,24,88,120,155])
set1 &= set2
li=list(set1)
print li
```

#--#

With a given list [12,24,35,24,88,120,155,88,120,155], write a program to print this list after removing all duplicate values with original order reserved.

Hints:
Use set() to store a number of values without duplicate.

Solution:

```
def removeDuplicate( li ):
   newli=[]
   seen = set()
   for item in li:
      if item not in seen:
         seen.add( item )
         newli.append(item)
```

```
    return newli

li=[12,24,35,24,88,120,155,88,120,155]
print removeDuplicate(li)
```

#--#
Question:

Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class.

Hints:
Use Subclass(Parentclass) to define a child class.

Solution:

```
class Person(object):
    def getGender( self ):
        return "Unknown"

class Male( Person ):
    def getGender( self ):
        return "Male"

class Female( Person ):
    def getGender( self ):
        return "Female"

aMale = Male()
aFemale= Female()
print aMale.getGender()
print aFemale.getGender()
```

#--#
Question:

Please write a program which count and print the numbers of each character in a string input by console.

Example:
If the following string is given as input to the program:

abcdefgabc

Then, the output of the program should be:

a,2
c,2
b,2
e,1

d,1
g,1
f,1

Hints:
Use dict to store key/value pairs.
Use dict.get() method to lookup a key with default value.

Solution:

```
dic = {}
s=raw_input()
for s in s:
    dic[s] = dic.get(s,0)+1
print '\n'.join(['%s,%s' % (k, v) for k, v in dic.items()])
```

#--#

Question:

Please write a program which accepts a string from console and print it in reverse order.

Example:
If the following string is given as input to the program:

rise to vote sir

Then, the output of the program should be:

ris etov ot esir

Hints:
Use list[::-1] to iterate a list in a reverse order.

Solution:

```
s=raw_input()
s = s[::-1]
print s
```

#--#

Question:

Please write a program which accepts a string from console and print the characters that have even indexes.

Example:
If the following string is given as input to the program:

H1e2l3l4o5w6o7r8l9d

Then, the output of the program should be:

Helloworld

Hints:
Use list[::2] to iterate a list by step 2.

Solution:

```
s=raw_input()
s = s[::2]
print s
#----------------------------------------#
```

Question:

Please write a program which prints all permutations of [1,2,3]

Hints:
Use itertools.permutations() to get permutations of list.

Solution:

```
import itertools
print list(itertools.permutations([1,2,3]))

#----------------------------------------#
```

Question:

Write a program to solve a classic ancient Chinese puzzle:
We count 35 heads and 94 legs among the chickens and rabbits in a farm. How many rabbits and how many chickens do we have?

Hint:
Use for loop to iterate all possible solutions.

Solution:

```
def solve(numheads,numlegs):
    ns='No solutions!'
    for i in range(numheads+1):
        j=numheads-i
        if 2*i+4*j==numlegs:
            return i,j
    return ns,ns

numheads=35
numlegs=94
solutions=solve(numheads,numlegs)
print solutions
```

#-------------------------------------#