**This document contains python practice programming questions, divided into 3 levels.**

**Level 1: Basic**

**Level 2: Intermediate**

**Level 3: Difficult**

Q1. '''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.'''

l=[]

for i in range(2000,3200,1):

if i%7==0 and i%5!=0:

l.append(str(i))

print(','.join(l))

Q2. '''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'''

def factorial(x):

if x==1 or x==0:

return 1

return x\*factorial(x-1)

x=int(input("enter a number : "))

print(factorial(x))

Q3. '''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}'''

d=dict()

n=int(input("enter a number : "))

for i in range(1,n+1):

d[i]=i\*i

print(d)

Q4. '''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')

'''

values=input('enter list of numbers: ')

l=values.split(",")

t=tuple(l)

print (l)

print (t)

Q5. '''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.

'''

class InputOutString(object):

def \_\_init\_\_(self):

self.s = ""

def getString(self):

self.s = input("Enter a string :")

def printString(self):

print (self.s.upper())

obj=InputOutString()

obj.getString()

obj.printString()

Q6. '''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.'''

import math

C=50

H=30

D=int(input("Enter the values :"))

print(math.sqrt((2\*C\*D)/H))

Q7. '''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,...,Y-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]] '''

import numpy

w, h = int(input("enter no. of rows")),int(input("enter no. of columns"))

Matrix = [[0 for x in range(w)] for y in range(h)]

for i in range(w):

for j in range(h):

Matrix[j][i]=i\*j

print(numpy.transpose(Matrix))

Q8. '''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'''

s = input("Enter : ")

words = [word for word in s.split(" ")]

print (" ".join(sorted(list(set(words)))))

Q9. '''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'''

lines = []

while True:

s = input("Start entering lines")

if s:

lines.append(s.upper())

else:

break;

for sentence in lines:

print (sentence)

Q10. '''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'''

s = input("Enter : ")

words = [word for word in s.split(" ")]

print (" ".join(sorted(list(set(words)))))

Q11. '''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'''

value = []

items=[x for x in input("Enter : ").split(',')]

for p in items:

intp = int(p, 2)

if not intp%5:

value.append(p)

print (','.join(value))

Q12. '''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.'''

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))

Q13. '''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'''

n=input("enter : ")

d={"DIGITS":0, "LETTERS":0}

for c in n:

if c.isdigit():

d['DIGITS']+=1

elif c.isalpha():

d['LETTERS']+=1

else:

pass

print(d)

Q14. '''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'''

s = input("Enter : ")

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(d)

Q15. '''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'''

a=input("Enter : ")

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)

Q16. '''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'''

values = input("Enter : ")

numbers = [x for x in values.split(",") if int(x)%2!=0]

print (",".join(numbers))

Q17. '''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'''

netAmount = 0

while True:

s = input("Enter in format D/W Amount : ")

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)

Q18. '''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]
3. At least 1 letter between [A-Z]
4. At least 1 character from [$#@]
5. Minimum length of transaction password: 6
6. 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'''

import re

value = []

items=[x for x in input("Enter : ").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))

Q 18. '''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')]'''

from operator import itemgetter, attrgetter

l = []

while True:

s = input("Enter comma seperated value : ")

if not s:

break

l.append(tuple(s.split(",")))

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

Q20. '''Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n'''

n = int(input())

divBy7 = [i for i in range(0, n) if (i % 7 == 0)]

print(divBy7)

def divChecker(n):

for i in range(n):

if i % 7 == 0:

value = True

else:

value = False

print(i, value)

divChecker(n)

Q21. '''Define a function that can receive two integral numbers in string form and compute their sum and then print it in console.'''

def funcalc(s1, s2):

return int(s1)+int(s2)

print(funcalc('4','3'))

Q22. '''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 all strings line by line.'''

def fun\_str(s1,s2):

if len(s1)>len(s2):

print(s1)

elif len(s2)>len(s1):

print(s2)

else:

print(s1)

print(s2)

s1=input("enter string 1: ")

s2=input("enter string 2: ")

fun\_str(s1,s2)

Q23. '''Define a function which can print a dictionary where the keys are numbers between 1 to n (both included) and the values are square of keys.'''

def squareddict(n):

d=dict()

for i in range(1,n+1):

d[i]=i\*i

print(d) #prints key value pair

for k in d.keys():

print (k) #prints only keys

for (k,v) in d.items(): #prints values not keys

print (v)

squareddict(8)

Q24.

1. print 1st five element => list[:5]
2. print last five elements => list[-5:]
3. print all but 1st five => list[5:]

Q25. '''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'''

import math

pos = [0,0]

while True:

s = input("Enter values : ")

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))))

Q26. '''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'''

freq = {} # frequency of words in text

line = input("Enter the line : ")

for word in line.split():

freq[word] = freq.get(word,0)+1

for key in sorted(freq):

print ("%s: %s" % (key, freq[key]))

Q27. '''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'''

class Person:

name='person'

def \_\_init\_\_(self,name=None):

self.name=name

pawan=Person("Pawan")

print ("%s name is %s" % (Person.name, pawan.name))

kumar=Person()

kumar.name='Kumar'

print ("%s name is %s" % (Person.name, kumar.name))

Q 28. '''Define a function which can generate and print a tuple where the value are square of numbers between 1 and n (both included).'''

def SquareTuple(n):

li=list()

for i in range(1,n+1):

li.append(i\*\*2)

print(li)

SquareTuple(20)

def SquareList(n):

li=[]

for i in range(1,n+1):

li.append(i\*\*2)

print(li)

SquareList(20)

def SquareDict(n):

d=dict()

for i in range(1,n+1):

d[i]=i\*\*2

print(d)

SquareDict(20)

Q 29. '''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).'''

tp=(1,2,3,4,5,6,7,8,9,10)

li=list()

for i in range(len(tp)):

if tp[i]%2==0:

li.append(tp[i])

tup=tuple(li)

print(tup)

Q 31. '''Define a class named American which has a static method called printNationality.

Use @staticmethod decorator to define class static method.'''

class American(object):

@staticmethod

def printNationality():

print ("America")

am = American()

am.printNationality()

American.printNationality()

Q 32. '''Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.

Use def methodName(self) to define a method.'''

class Circle(object):

def \_\_init\_\_(self,r):

self.radius=r

def area(self):

return self.radius \* 2 \* 3.14

ac=Circle(12)

print(ac.area())