Question 1:

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

# Python Program to calculate the square root

import math

# Note: change this value for a different result

C=50

H=30

I=[]

mult=1

D=[100,150,180]

for i in D:

mult=round((2\*i\*mult\*C/H)\*\*0.5)

I.append(mult)

print(I)

Question 2:

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]]

A: from numpy import fromfunction as ff

x,y = (3,5)

arr = ff(lambda x,y:x\*y,(x,y))

print(arr)

Question 3:

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

Question 4:

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

A: l="hello world and practice makes perfect and hello world again"

l=l.split()

print("original list",l)

res = [\*set(l)]

res1=res.sort()

print("List after removing duplicate elements: ", res)

print("List after arranging elements alphanumerically : ", res)

Question 5:

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

test\_str="hello world! 123"

#Splitting text and number in string

text=""

numbers=""

res=[]

for i in test\_str:

if(i.isdigit()):

numbers+=i

else:

text+=i

res.append(text)

res.append(numbers)

print("The tuple after the split of string and number : " + str(res))

l1=res[0]

l2=res[1]

res1 = len([ele for ele in l1 if ele.isalpha()])

print('count of alphabets: {}'.format(res1))

res2 = len([ele for ele in l2 if ele.isdigit()])

print('count of digits: {}'.format(res2))

Question 6:

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

A: s=input("enter sequence of comma separated passwords :")

l, u, p, d = 0, 0, 0, 0

if (6<= len(s) <= 12):

for i in s:

# counting lowercase alphabets

if (i.islower()):

l+=1

# counting uppercase alphabets

if (i.isupper()):

u+=1

# counting digits

if (i.isdigit()):

d+=1

# counting the mentioned special characters

if(i=='@'or i=='$' or i=='\_'):

p+=1

if (l>=1 and u>=1 and p>=1 and d>=1 and l+p+u+d==len(s)):

print("Valid Password")

else:

print("Invalid Password")