**ASSIGNMENT-13**

**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

In [1]:

**from** math **import** sqrt

**def** calculateProgram():

in\_num **=** eval(input("Enter the Input: "))

out\_num **=** []

C **=** 50 *# Declaring and initializing constant C*

H **=** 30 *# Declaring and initializing constant H*

**for** ele **in** in\_num:

Q **=** str(int(sqrt((2**\***C**\***ele)**/**H)))

out\_num**.**append(Q)

print("Output: {}"**.**format(','**.**join(out\_num)))

calculateProgram()

Enter the Input: 100,150,180

Output: 18,22,24

**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]]

In [2]:

**import** array **as** arr

**def** generateArray():

in\_x **=** int(input('Enter the No of Rows:'))

in\_y **=** int(input('Enter the No of Columns:'))

out\_array **=** []

**for** ele **in** range(in\_x):

out\_array**.**insert(in\_x,[])

**for** sub\_ele **in** range(in\_y):

out\_array[ele]**.**append(ele**\***sub\_ele)

print(out\_array)

generateArray()

Enter the No of Rows:3

Enter the No of Columns:5

[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

**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

In [3]:

**def** sortString():

in\_string **=** input("Enter the Input String: ")

out\_string **=** ','**.**join(sorted(in\_string**.**split(',')))

print(f'Output: {out\_string}')

sortString()

Enter the Input String: without,hello,bag,world

Output: bag,hello,without,world

**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

In [4]:

**def** sortAlphaNumerically():

in\_string **=** input("Enter the Input String: ")

out\_string **=** ' '**.**join(sorted(sorted(list(set(in\_string**.**split(" "))))))

print(f'Output: {out\_string}')

sortAlphaNumerically()

Enter the Input String: hello world and practice makes perfect and hello world again

Output: again and hello makes perfect practice world

**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

In [5]:

**def** countLetterAndDigits():

in\_string **=** input("Enter the Input String: ")

lettersList **=** 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'

digitsList **=** '0123456789'

letters **=** 0

digits **=** 0

**for** ele **in** in\_string:

**if** ele **in** lettersList:

letters **+=** 1

**if** ele **in** digitsList:

digits **+=** 1

print(f'LETTERS {letters} \nDIGITS {digits}')

countLetterAndDigits()

Enter the Input String: hello world! 123

LETTERS 10

DIGITS 3

**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]
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

In [6]:

**def** checkPassword():

in\_string **=** input("Enter the Input String: ")

small\_list **=** "abcdefghijklmnopqrstuvwxyz"

cap\_list **=** "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

num\_list **=** "0123456789"

special\_list **=** "$#@"

**for** ele **in** in\_string**.**split(","):

**if** len(ele) **<=** 12 **and** len(ele) **>=**6 :

**if** any(i**.**isupper() **for** i **in** ele):

**if** any(i**.**islower() **for** i **in** ele):

**if** any(i **for** i **in** ele **if** i **in** special\_list):

print(ele)

checkPassword()

Enter the Input String: ABd1234@1,a F1#,2w3E\*,2We3345

ABd1234@1