**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**

import math

C = 50

H = 30

def calculate\_Q(D):

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

input\_str = input("Enter comma-separated values of D: ")

D\_values = [int(val) for val in input\_str.split(",")]

Q\_values = [calculate\_Q(D) for D in D\_values]

print(Q\_values)

**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]]**

def generate\_array(X, Y):

return [[i \* j for j in range(Y)] for i in range(X)]

input\_str = input("Enter X, Y: ")

X, Y = [int(val) for val in input\_str.split(",")]

result = generate\_array(X, Y)

for row in result:

print(row)

**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**

def sort\_words(words):

return sorted(words)

input\_str = input("Enter comma-separated words: ")

words = input\_str.split(",")

result = sort\_words(words)

print(", ".join(result))

**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**

def sort\_and\_remove\_duplicates(words):

words = sorted(list(set(words)))

return words

input\_str = input("Enter whitespace-separated words: ")

words = input\_str.split()

result = sort\_and\_remove\_duplicates(words)

print(" ".join(result))

**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**

def count\_letters\_and\_digits(sentence):

letters = 0

digits = 0

for char in sentence:

if char.isdigit():

digits += 1

elif char.isalpha():

letters += 1

return letters, digits

input\_str = input("Enter a sentence: ")

letters, digits = count\_letters\_and\_digits(input\_str)

print("LETTERS", letters)

print("DIGITS", digits)

**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**

def is\_valid\_password(password):

if len(password) < 6 or len(password) > 12:

return False

has\_lowercase = False

has\_uppercase = False

has\_digit = False

has\_special = False

for char in password:

if char.isdigit():

has\_digit = True

elif char.islower():

has\_lowercase = True

elif char.isupper():

has\_uppercase = True

elif char in "$#@":

has\_special = True

return has\_lowercase and has\_uppercase and has\_digit and has\_special

input\_str = input("Enter comma-separated passwords: ")

passwords = input\_str.split(",")

valid\_passwords = []

for password in passwords:

if is\_valid\_password(password):

valid\_passwords.append(password)

if valid\_passwords:

print(", ".join(valid\_passwords))

else:

print("No valid passwords found.")