|  |
| --- |
| **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

# define the fixed values of C and H

C = 50

H = 30

# take input of comma-separated values of D

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

D\_list = input\_str.split(",")

# convert the strings to floats and calculate Q for each value of D

Q\_list = []

for D\_str in D\_list:

D = float(D\_str)

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

Q\_list.append(round(Q))

# print the output as comma-separated values

output\_str = ",".join(str(Q) for Q in Q\_list)

print(output\_str)

|  |
| --- |
| **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]]** |
| # take input of X and Y  input\_str = input("Enter values of X and Y (comma-separated): ")  X, Y = map(int, input\_str.split(","))  # create the 2-dimensional array  array\_2d = []  for i in range(X):  row = []  for j in range(Y):  row.append(i\*j)  array\_2d.append(row)  # print the output as a formatted string  output\_str = "\n".join(str(row) for row in array\_2d)  print(output\_str) |
|  |

**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**

# take input of comma-separated words

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

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

# sort the words alphabetically

words\_sorted = sorted(words)

# print the output as a comma-separated string

output\_str = ",".join(words\_sorted)

print(output\_str)

**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**

# take input of whitespace-separated words

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

words = input\_str.split()

# remove duplicate words

words\_unique = list(set(words))

# sort the remaining words alphanumerically

words\_sorted = sorted(words\_unique)

# print the output as a whitespace-separated string

output\_str = " ".join(words\_sorted)

print(output\_str)

**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**

# take input of a sentence

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

# initialize counters for letters and digits

num\_letters = 0

num\_digits = 0

# count the number of letters and digits

for char in input\_str:

if char.isalpha():

num\_letters += 1

elif char.isdigit():

num\_digits += 1

# print the output

print("LETTERS", num\_letters)

print("DIGITS", num\_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**

import re

# take input of comma-separated passwords

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

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

# define the regex pattern for valid passwords

pattern = r"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$#])[A-Za-z\d@$#]{6,12}$"

# loop over the passwords and check their validity

valid\_passwords = []

for password in passwords:

if re.match(pattern, password):

valid\_passwords.append(password)

# print the output as comma-separated string of valid passwords

output\_str = ",".join(valid\_passwords)

print(output\_str)