|  |
| --- |
| **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

def calculate\_values(d\_values):

result = []

c = 50

h = 30

d\_list = d\_values.split(',')

for d in d\_list:

q = math.sqrt((2 \* c \* int(d)) / h)

result.append(str(round(q)))

return ','.join(result)

# Test the function with the example values

input\_sequence = "100,150,180"

output\_sequence = calculate\_values(input\_sequence)

print(output\_sequence)

|  |  |
| --- | --- |
| **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\_2d\_array(x, y):  array\_2d = []    for i in range(x):  row = []  for j in range(y):  row.append(i \* j)  array\_2d.append(row)    return array\_2d  # Test the function with the example inputs  x = 3  y = 5  output\_array = generate\_2d\_array(x, y)  print(output\_array) |

**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(sequence):

words = sequence.split(",")

words.sort()

sorted\_sequence = ",".join(words)

return sorted\_sequence

# Test the function with the example input

input\_sequence = "without,hello,bag,world"

output\_sequence = sort\_words(input\_sequence)

print(output\_sequence)

**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 process\_words(sequence):

words = sequence.split()

unique\_words = list(set(words))

sorted\_words = sorted(unique\_words)

result = " ".join(sorted\_words)

return result

# Test the function with the example input

input\_sequence = "hello world and practice makes perfect and hello world again"

output\_sequence = process\_words(input\_sequence)

print(output\_sequence)

**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):

letter\_count = 0

digit\_count = 0

for char in sentence:

if char.isalpha():

letter\_count += 1

elif char.isdigit():

digit\_count += 1

return letter\_count, digit\_count

# Test the function with the example input

input\_sentence = "hello world! 123"

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

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]** |
|  |

|  |
| --- |
| **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

def check\_passwords(passwords):

valid\_passwords = []

for password in passwords:

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

continue

if not re.search("[a-z]", password):

continue

if not re.search("[A-Z]", password):

continue

if not re.search("[0-9]", password):

continue

if not re.search("[$#@]", password):

continue

valid\_passwords.append(password)

return valid\_passwords

# Test the function with the example input

input\_passwords = "ABd1234@1,a F1#,2w3E\*,2We3345"

password\_list = input\_passwords.split(",")

valid\_passwords = check\_passwords(password\_list)

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

print(output)