**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\_formula(args):

    c = 50

    h = 30

    result = []

    for d in args:

        result.append(int(math.sqrt((2\*c\*int(d))/h)))

    print(result)

inp = list(input("Enter comma separated numbers input: ").split(','))

calculate\_formula(inp)

Enter comma separated numbers input: 100,150,180

[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.**

x = int(input("Enter X value: "))

y = int(input("Enter Y value: "))

matrix = []

for i in range(x):

    row = []

    for j in range(y):

        row.append(i\*j)

    matrix.append(row)

print(matrix)

Enter X value: 3

Enter Y value: 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.**

words = list(input("Enter comma seperated words: ").split(','))

words.sort()

print(','.join(words))

Enter comma seperated words: without,hello,bag,world

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.**

words = list(set(input("Enter comma seperated words: ").split(' ')))

words.sort()

print(' '.join(words))

Enter comma seperated words: hello world and practice makes perfect and hello world again

again and hello makes perfect practice world

**5. Write a program that accepts a sentence and calculate the number of letters and digits.**

words = input("Enter a sentence: ")

letters = 0

digits = 0

for c in words:

    if (ord(c) >= ord('a') and ord(c) <= ord('z')) or (ord(c) >= ord('A') and ord(c) <= ord('Z')):

        letters += 1

    elif ord(c) >= ord('0') and ord(c) <= ord('9'):

        digits += 1

print("LETTERS {}".format(letters))

print("DIGITS {}".format(digits))

Enter a sentence: 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.**

def check\_smallcase(password):

    check = False

    for p in password:

        if ord(p) >= ord('a') and ord(p) <= ord('z'):

            check = True

            break

    return check

def check\_uppercase(password):

    check = False

    for p in password:

        if ord(p) >= ord('A') and ord(p) <= ord('Z'):

            check = True

            break

    return check

def check\_digit(password):

    check = False

    for p in password:

        if ord(p) >= ord('1') and ord(p) <= ord('9'):

            check = True

            break

    return check

def check\_specialchars(password):

    check = False

    specials = "!@#$%^&\*()\_+=->;,:/?`"

    for p in password:

        if p in specials:

            check = True

            break

    return check

def check\_minlength(password):

    return len(password) >= 6

def check\_maxlength(password):

    return len(password) <= 12

passwords = list(set(input("Enter comma seperated passwords: ").split(',')))

valid\_passwords = []

for password in passwords:

    if check\_smallcase(password) and check\_uppercase(password) and check\_digit(password) and check\_specialchars(password) and check\_minlength(password) and check\_maxlength(password):

        valid\_passwords.append(password)

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

Enter comma seperated passwords: ABd1234@1,a F1#,2w3E\*,2We3345, EKd9876@1,a F1#,2w3E\*,2We3345

EKd9876@1,ABd1234@1