|  |
| --- |
| 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

def calc(C,D,H):

x = [] #list to store operated values

for i in D:

y = int((2\*C\*int(i)/H)\*\*.5)

x.append(y)

return x

C = 50

H = 30

D = input('enter the values separated by comma ').split(',')

R = calc(C,D,H)

for i,x in enumerate(R): #code block to give output in desired format

if i==len(R)-1:

print(x)

else:

print (x,end=',')

|  |
| --- |
| 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 array2d(row,col):

from numpy import array

arr = array([[i\*j for i in range(col)] for j in range(row)])

return arr

X= int(input('enter number of rows '))

Y= int(input('enter number of colums '))

print (array2d(X,Y))

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

words = input('enter words separated by comma ').split(',')

for i in sorted(words):

if words.index(i)==len(words)-1: #we don’t want to print comma after last word

print(i)

else:

print(i,end=',')

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\_uniq(words):

uniq = set() #set to store unique words

words2= words.copy() #dummy list words2 to perform remove function on

for i in words:

if i in uniq:

words2.remove(i) #if a word already exist, remove the duplicate

else:

uniq.add(i) #add the unique words to the set

return sorted(words2)

words = input('enter words separated by whitespace: ').split(' ')

for i in sort\_uniq(words):

print(i,end=' ')

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\_char(string):

az = 0 #counter for alphabets

num = 0 #counter for digits

for i in string:

if i.isalpha() : #checking if a character is alphabet and then increasing counter

az+=1

if i.isdigit():

num+=1

return az,num

s1 = input('enter the string: ')

a,n = count\_char(s1)

print('LETTERS ',a)

print('DIGITS ',n)

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

from re import search

def check\_pass(passwords):

passed = []

for i in passwords:

if bool(search('^(?=.\*[a-z])(?=.\*[0-9])(?=.\*[A-Z])(?=.\*[$#@]){6,12}.\*$',i)): #regex combined to check

passed.append(i)

return passed

x = input('enter passwords separated by comma: ').split(',')

y = check\_pass(x)

for i in y :

if y.index(i)==len(y)-1:

print(i)

else:

print(i,end=',')