|  |
| --- |
| 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

value = []

items=[x for x in input().split(',')]

for d in items:

value.append(str(int(round(math.sqrt(2\*c\*float(d)/h)))))

print(','.join(value))

|  |
| --- |
| 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]] |
|  |

x = int(input("Enter the number."))

y = int(input("Enter the number."))

matrix = []

for i in range(x):

row = []

for j in range(y):

value = i \* j

row.append(value)

matrix.append(row)

print(matrix)

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

input\_str = input("Enter sting with comma separatedd sequence: ")

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

words.sort()

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

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

input\_str = input("Enter the string with space seperated : ")

words = input\_str.split()

words = list(set(words))

words.sort()

output\_str = " ".join(words)

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

sentence = input("Enter sentance with mix with digit : ")

num\_letters = 0

num\_digits = 0

for char in sentence:

if char.isalpha():

num\_letters += 1

elif char.isdigit():

num\_digits += 1

print("Number of letters:", num\_letters)

print("Number of 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

def check\_password(password):

# At least 1 letter between [a-z]

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

return False

# At least 1 number between [0-9]

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

return False

# At least 1 letter between [A-Z]

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

return False

# At least 1 character from [$#@]

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

return False

# Minimum length of transaction password: 6

if len(password) < 6:

return False

# Maximum length of transaction password: 12

if len(password) > 12:

return False

# If the password passes all checks, it is considered valid

return True

def main():

# Prompt the user to input a sequence of comma separated passwords

passwords = input("Enter a sequence of comma separated passwords: ")

# Split the input into a list of passwords

passwords = passwords.split(',')

# Loop through each password in the list and check its validity

valid\_passwords = []

for password in passwords:

if check\_password(password):

valid\_passwords.append(password)

# Print the valid passwords, each separated by a comma

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

if \_\_name\_\_ == '\_\_main\_\_':

main()