| 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

inp = input().split(',')

for D in inp:

C = 50

H = 30

Q = ((2\*C\*int(D))/H)\*\*0.5

print(Q,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]] |
| --- |
|  |

X,Y = list(map(int,input().split(',')))

output = []

for x in range(X):

row = []

for y in range(Y):

row.append(x\*y)

output.append(row)

print(output)

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

inp = input().split(',')

print(','.join(sorted(inp)))

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

inp = input().split()

unq\_inp = []

for x in inp:

if x not in unq\_inp:

unq\_inp.append(x)

print(\*sorted(unq\_inp))

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

inp = input()

digit = 0

letters = 0

for x in inp:

if x.isdigit():

digit+=1

elif x.isalpha():

letters+=1

print('LETTERS', letters)

print('DIGITS', digit)

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

**Ans.**

**inp = input().split(',')**

**cap = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'**

**for x in inp:**

**if all([len(x)>5,len(x)<13,any([y in '&#@' for y in x]),any([y.isalpha() for y in x]),any([y.isdigit() for y in x]),any([y in cap for y in x])]):**

**print(x)**