|  |
| --- |
| 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

**Ans: 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]] |
|  |

**Ans: row\_num = int(input("enter value for X: "))**

**col\_num = int(input("enter value for Y: "))**

**multi\_list = [[0 for col in range(col\_num)] for row in range(row\_num)]**

**for row in range(row\_num):**

**for col in range(col\_num):**

**multi\_list[row][col]= row\*col**

**print(multi\_list)**

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

**Ans: phrase = input("Input words: ")**

**phrase\_list = phrase.split(",")**

**phrase\_list.sort()**

**print((', ').join(phrase\_list))**

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

**Ans: phrase = input("Type in: ")**

**phrase\_splited = phrase.split(' ')**

**word\_list = []**

**for i in phrase\_splited:**

**if i not in word\_list:**

**word\_list.append(i)**

**else:**

**continue**

**word\_list.sort()**

**print((' ').join(word\_list))**

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

**Ans: phrase = input("Type in: ")**

**phrase = list(phrase)**

**l, d = 0, 0**

**for i in phrase:**

**if i.isalpha():**

**l = l + 1**

**if i.isdigit():**

**d = d + 1**

**else:**

**pass**

**print("Letters:", l)**

**print("Digits:", d)**

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: import re**

**passwords = input("Type in: ")**

**passwords = passwords.split(",")**

**accepted\_pass = []**

**for i in passwords:**

**if len(i) < 6 or len(i) > 12:**

**continue**

**elif not re.search("([a-z])+", i):**

**continue**

**elif not re.search("([A-Z])+", i):**

**continue**

**elif not re.search("([0-9])+", i):**

**continue**

**elif not re.search("([!@$%^&])+", i):**

**continue**

**else:**

**accepted\_pass.append(i)**

**print((" ").join(accepted\_pass))**