|  |
| --- |
| **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**

**numbers = input("Provide D: ")**

**numbers = numbers.split(',')**

**result\_list = []**

**for D in numbers:**

**Q = round(math.sqrt(2 \* 50 \* int(D) / 30))**

**result\_list.append(Q)**

**print(result\_list)**

|  |
| --- |
| **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:-**

**userInput = raw\_input("Enter values for row and column number:\t").split(",")**

**for x in range(userInput):**

**for y in range(userInput[x]):**

**userInput[x][y]=x\*y**

**print(userInput[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**

**ANS:-**

**items = input("Input comma separated sequence of words")**

**words = [word for word in items.split(",")]**

**print(",".join(sorted(list(set(words)))))**

**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:-**

**from collections import Counter**

**def remov\_duplicates(input):**

**# split input string separated by space**

**input = input.split(" ")**

**# now create dictionary using counter method**

**# which will have strings as key and their**

**# frequencies as value**

**UniqW = Counter(input)**

**# joins two adjacent elements in iterable way**

**s = " ".join(UniqW.keys())**

**print (s)**

**# Driver program**

**if \_\_name\_\_ == "\_\_main\_\_":**

**input = 'Python is great and Java is also great'**

**remov\_duplicates(input)**

**Output**

and great Java Python is also

**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:-**

**# define all digits as string**

**all\_digits = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']**

**# define all letters**

**all\_letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l','m', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']**

**# given string**

**string = "geeks2for3geeks"**

**# initialized value**

**total\_digits = 0**

**total\_letters = 0**

**# iterate through all characters**

**for s in string:**

**# if character found in all\_digits then increment total\_digits by one**

**if s in all\_digits:**

**total\_digits += 1**

**# if character found in all\_letters then increment total\_letters by one**

**elif s in all\_letters:**

**total\_letters += 1**

**print("Total letters found :-", total\_letters)**

**print("Total digits found :-", total\_digits)**

**Output:**

Total letters found :- 13

Total digits found :- 2

**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:-**

**# Python program to check validation of password**

**# Module of regular expression is used with search()**

**import re**

**password = "R@m@\_f0rtu9e$"**

**flag = 0**

**while True:**

**if (len(password)<=8):**

**flag = -1**

**break**

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

**flag = -1**

**break**

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

**flag = -1**

**break**

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

**flag = -1**

**break**

**elif not re.search("[\_@$]" , password):**

**flag = -1**

**break**

**elif re.search("\s" , password):**

**flag = -1**

**break**

**else:**

**flag = 0**

**print("Valid Password")**

**break**

**if flag == -1:**

**print("Not a Valid Password ")**

**Output**

Valid Password