|  |
| --- |
| 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:- def calculate(N, C, H):**

**for D in N:**

**result\_1 = (2 \* C \* D) / H**

**result = result\_1 \*\* 0.5**

**print(int(result)) # converting result to integer for whole number output**

**calculate((100, 150, 180), 50, 30)**

|  |
| --- |
| 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:- def generate\_array(X, Y):**

**array = []**

**for i in range(X):**

**row = []**

**for j in range(Y):**

**row.append(i \* j)**

**array.append(row)**

**return array**

**# Input values**

**X = 3**

**Y = 5**

**# Generate the 2-dimensional array**

**output\_array = generate\_array(X, Y)**

**# Print the generated array**

**print(output\_array)**

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:- def sorting(\*args):**

**sorted\_words = sorted(args)**

**result = ",".join(sorted\_words)**

**print(result)**

**sorting("without", "hello", "bag", "world")**

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:- def duplicate(a):**

**result\_1 = a.split()**

**result\_2 = list(set(result\_1))**

**result\_3 = sorted(result\_2)**

**result = " ".join(result\_3)**

**print(result)**

**duplicate("hello world and practice makes perfect and hello world again")**

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:- def split\_letters(string):**

**letter = 0**

**digit = 0**

**letters = [char for char in string]**

**for i in letters:**

**if i.isalpha():**

**letter += 1**

**elif i.isdigit():**

**digit += 1**

**else:**

**continue**

**print("LETTERS", letter)**

**print("DIGITS", digit)**

**split\_letters("Hello World! 123")**

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:- def check\_password(password):**

**valid\_passwords = []**

**passwords = password.split(',')**

**for password in passwords:**

**if 6 <= len(password) <= 12:**

**if any(char.islower() for char in password):**

**if any(char.isupper() for char in password):**

**if any(char.isdigit() for char in password):**

**if any(char in ['$','@','#'] for char in password):**

**valid\_passwords.append(password)**

**valid\_passwords\_str = ', '.join(valid\_passwords)**

**print(valid\_passwords\_str)**

**check\_password("ABd1234@1,a F1#,2w3E\*,2We3345")**