|  |
| --- |
| 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 in with comma separated: ")

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

result\_list **=** []

result\_string **=** ''

**for** D **in** numbers:

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

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

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

Provide D in with comma separated: 2,39,45

3,11,12

|  |
| --- |
| 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** createMatrix(n,m):    M**=**[]  print("Enter the element :")  **for** i **in** range(n):  *#stor row*  row **=**[]  **for** j **in** range(m):  row**.**append(i**\***j)  M**.**append(row)  **return**(M)  x **=** int(input("Enter x : "))  y **=** int(input("enter y : "))  createMatrix(x,y)  Enter x : 3  enter y : 5  Enter the element :  Out[5]:  [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]] |
|  |

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**=**[x **for** x **in** input('Enter comma seperated words ')**.**split(',')]

items**.**sort()

print(','**.**join(items))

Enter comma seperated words without,hello,bag,world

bag,hello,without,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: items**=**[x **for** x **in** input('Enter space sepeated words ')**.**split(' ')]

print(' '**.**join(sorted(list(set(items)))))

Enter space sepeated words hello world and practice makes perfect and hello world again

again and hello makes perfect practice world

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: s **=** input("Input a string : ")

digits**=**letters**=**0

**for** c **in** s:

**if** c**.**isdigit():

digits **+=** 1

**elif** c**.**isalpha():

letters **+=** 1

**else**:

**pass**

print("Letters", letters)

print("Digits", digits)

Input a string : akashdeep364

Letters 9

Digits 3

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

password**=** input("Enter your password : ")

x **=** **True**

**while** x:

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

**break**

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

**break**

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

**break**

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

**break**

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

**break**

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

**break**

**else**:

print("Valid Password")

x**=False**

**break**

**if** x:

print("Not a Valid Password")

Enter your password : 123#dldjj

Not a Valid Password