**ADITYA AMIN**

**ASSIGN : 13**

|  |
| --- |
| 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

import math

# Fixed values of C and H

C = 50

H = 30

def calculate\_Q(D):

Q\_values = []

for d in D:

# Calculate Q using the formula

q = int(math.sqrt((2 \* C \* int(d)) / H))

Q\_values.append(q)

return Q\_values

# Input values for D as comma-separated string

input\_str = "100,150,180"

# Convert input string to list of integers

D = list(map(int, input\_str.split(',')))

# Calculate Q values

Q\_values = calculate\_Q(D)

# Convert Q values to comma-separated string

output\_str = ','.join(map(str, Q\_values))

# Print the calculated Q values

print("Output: ", output\_str)

|  |
| --- |
| 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]] |
|  |

def generate\_2d\_array(X, Y):

arr = []

for i in range(X):

row = []

for j in range(Y):

row.append(i \* j)

arr.append(row)

return arr

# Take input for X and Y as comma-separated string

input\_str = input("Enter X, Y: ")

# Convert input string to integers

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

# Generate 2-dimensional array

arr = generate\_2d\_array(X, Y)

# Print the generated array

print("Output:")

for row in arr:

print(row)

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

def sort\_words(words):

"""

Sorts the words alphabetically.

Args:

words (list): List of words.

Returns:

list: Sorted list of words.

"""

words.sort() # Sort the words alphabetically

return words

# Take input as comma-separated string

input\_str = input("Enter comma-separated words: ")

# Split input string by comma to get a list of words

words = input\_str.split(',')

# Sort the words alphabetically

sorted\_words = sort\_words(words)

# Join the sorted words using comma as separator

output\_str = ",".join(sorted\_words)

# Print the sorted words

print("Output:")

print(output\_str)

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

def remove\_duplicates(words):

return list(set(words)) # Convert list to set to remove duplicates, then back to list

def sort\_words(words):

words.sort() # Sort the words alphanumerically

return words

# Take input as whitespace-separated string

input\_str = input("Enter whitespace-separated words: ")

# Split input string by whitespace to get a list of words

words = input\_str.split()

# Remove duplicate words

unique\_words = remove\_duplicates(words)

# Sort the unique words alphanumerically

sorted\_words = sort\_words(unique\_words)

# Join the sorted words using whitespace as separator

output\_str = " ".join(sorted\_words)

# Print the sorted words

print("Output:")

print(output\_str)

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

def count\_letters\_and\_digits(sentence):

letters = 0

digits = 0

# Iterate through each character in the sentence

for char in sentence:

if char.isalpha(): # Check if character is a letter

letters += 1

elif char.isdigit(): # Check if character is a digit

digits += 1

return letters, digits

# Take input as a sentence

input\_sentence = input("Enter a sentence: ")

# Call the function to count letters and digits

letters, digits = count\_letters\_and\_digits(input\_sentence)

# Print the results

print("LETTERS", letters)

print("DIGITS", digits)

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

import re

def validate\_password(password):

# Check if password has at least 1 lowercase letter

if not re.search(r'[a-z]', password):

return False

# Check if password has at least 1 uppercase letter

if not re.search(r'[A-Z]', password):

return False

# Check if password has at least 1 digit

if not re.search(r'[0-9]', password):

return False

# Check if password has at least 1 special character [$#@]

if not re.search(r'[$#@]', password):

return False

# Check if password has minimum length of 6 and maximum length of 12

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

return False

return True

# Take input as comma separated passwords

input\_passwords = input("Enter passwords separated by commas: ")

# Split input passwords by comma to create a list of passwords

passwords = input\_passwords.split(',')

# Iterate through each password and check for validity

valid\_passwords = []

for password in passwords:

if validate\_password(password):

valid\_passwords.append(password)

# Join valid passwords with commas and print the result

print("Valid passwords: ", ', '.join(valid\_passwords))