**Python programs day-1** Date: 21/02/2024

*1.Write a Python program to print the following pattern.*

**Program:**

char = input("Enter the Character to be printed: ")

num\_rows = int(input("Number of rows: "))

for i in range(1, num\_rows + 1):

for j in range(1, i + 1):

print(char, end=" ")

print()

**Sample Input:**

Enter the Character to be printed: 5

Number of rows.: 5

**Output:**

5

5 5

5 5 5

5 5 5 5

5 5 5 5 5

*2. What is the output of the following Program? (HINT: Answer is NOT 2)*

**Program:**

def my\_func(x):

if x<0:

return 1

return x \* my\_func(x-1)

x=int(input())

print(my\_func(x))

*3. Find the LSD & MSD of the given number*

**Program:**

def find\_MSD\_LSD(num):

num\_str = str(num)

if len(num\_str) == 1:

return f"MSD: {num\_str}, LSD: {num\_str}"

MSD = next((digit for digit in num\_str if digit != '0'), '0')

LSD = num\_str[-1]

return f"MSD: {MSD}, LSD: {LSD}"

decimal\_num = input("Enter a decimal number: ")

if decimal\_num.isdigit():

decimal\_num = int(decimal\_num)

print(find\_MSD\_LSD(decimal\_num))

else:

print("Invalid input. Please enter a valid decimal number.")

Input: 1010101

Output:

MSD:1

LSD:1

*4. Program to remove duplicates from the sorted array*

**Program:**

def remove\_duplicates\_and\_sort(arr):

unique\_arr = []

for num in arr:

if num not in unique\_arr:

unique\_arr.append(num)

unique\_arr.sort()

return unique\_arr

input\_list = [15,14,25,14,32,14,31]

result = remove\_duplicates\_and\_sort(input\_list)

print(result)

**Input:** Array = {15, 14, 25, 14, 32, 14, 31}

**Output:** Sorted Array = {14, 15, 25, 31, 32}

*5.Write a program to print the number of vowels in the given statement?*

**Program:**

def count\_vowels(statement):

vowels = {'a', 'e', 'i', 'o', 'u'}

vowel\_count = 0

for char in statement:

if char.lower() in vowels:

vowel\_count += 1

return vowel\_count

input\_statement = input("Enter a statement: ")

result = count\_vowels(input\_statement)

print(f"The number of vowels in the statement is: {result}")

**Output:**

Saveetha School of Engineering Sample

Number of vowels = 12

*6. Write a program to print unique permutations of a given number*

**Program:**

from itertools import permutations

def unique\_permutations(num):

num\_str = str(num)

unique\_perms = set(permutations(num\_str))

for perm in unique\_perms:

print(''.join(perm))

# Sample Input

num = 143

print(f"Given Number: {num}")

print("Permutations are:")

unique\_permutations(num)

**Output:**

134

143

314

341

413

431

*7. Write a program to reverse a word using loop? (Not to use inbuilt functions)*

**Program:**

def reverse\_word(word):

reversed\_word = ""

for char in word:

reversed\_word = char + reversed\_word

return reversed\_word

def reverse\_string(input\_str):

words = input\_str.split()

reversed\_words = [reverse\_word(word) for word in words]

reversed\_string = " ".join(reversed\_words)

return reversed\_string

# Example usage:

input\_string = "TEMPLE code in python"

reversed\_result = reverse\_string(input\_string)

print(f"Original string: {input\_string}\nReversed string: {reversed\_result}")

**Output:**

ELPMET

*8. Find the Cube and square of the given number:*

**Program:**

def square\_and\_cube(number):

square = number \*\* 2

cube = number \*\* 3

return square, cube

# Example usage:

number = int(input("Enter a number: "))

square, cube = square\_and\_cube(number)

print("Square of", number, "is:", square)

print("Cube of", number, "is:", cube)

**Output:**

Input: 2

Output: square: 4

Cube: 8

*9. Write a program to find the sum of digits of N digit number (sum should be single digit)*

**Program:**

def digit\_sum(n):

# Function to calculate the sum of digits

return sum(map(int, str(n)))

def single\_digit\_sum(num):

# Function to ensure the result is a single-digit sum

while num >= 10:

num = digit\_sum(num)

return num

# Example usage:

N = int(input("Enter N value: "))

number = int(input(f"Enter {N} digit number: "))

sum\_of\_digits = digit\_sum(number)

result = single\_digit\_sum(sum\_of\_digits)

print(f"Sum of digits: {sum\_of\_digits}\nSingle-digit sum: {result}")

**Sample Input:**

Enter N value : 3

Enter 3 digit number: 143

**Sample Output:**

Sum of 3 digit number: 8

*10. Write a program to arrange the letters of the word alphabetically in reverse order.(use build -in function)*

**Program**:

def arrange\_letters(word):

char\_list = list(word)

char\_list.sort(reverse=True)

arranged\_word = ''.join(char\_list)

return arranged\_word

input\_word = input("Enter a word: ")

result = arrange\_letters(input\_word)

print(f"The arranged word is: {result}")

**Sample Input:**

Enter the word : MOSQUE

**Sample Output:**

Alphabetical Order: U S Q O M E

*11. Find the LCM and GCD of n numbers?*

**Program:**

import math

def lcm(a, b):

# Calculate the least common multiple (LCM) of two numbers

return abs(a \* b) // math.gcd(a, b)

def gcd(a, b):

# Calculate the greatest common divisor (GCD) of two numbers

return math.gcd(a, b)

# Get input values

n = int(input("Enter the number of values: "))

# Initialize variables to store the input numbers and LCM/GCD

numbers = []

lcm\_val = None

gcd\_val = None

# Get input numbers and calculate LCM/GCD

for i in range(n):

num = int(input(f"Enter number {i+1}: "))

numbers.append(num)

if i == 0:

lcm\_val = num

gcd\_val = num

else:

lcm\_val = lcm(lcm\_val, num)

gcd\_val = gcd(gcd\_val, num)

# Print LCM and GCD

print(f"LCM: {lcm\_val}")

print(f"GCD: {gcd\_val}")

**Sample Input:**

N value = 2 Number 1 = 16

Number 2 = 20

**Sample Output**:

LCM = 80

*12. Write a program using choice to check*

*Case 1: Given string is palindrome or not*

*Case 2: Given number is palindrome or not*

**Program:**

import random

def check\_string\_palindrome():

input\_string = input("Enter a string: ")

if input\_string == input\_string[::-1]:

print("The given string is a palindrome.")

else:

print("The given string is not a palindrome.")

def check\_number\_palindrome():

input\_number = input("Enter a number: ")

if input\_number == input\_number[::-1]:

print("The given number is a palindrome.")

else:

print("The given number is not a palindrome.")

choice = random.choice([1, 2])

if choice == 1:

check\_string\_palindrome()

else:

check\_number\_palindrome()

**Sample Input:**

**Case = 1**

String = MADAM

**Sample Output:**

Palindrome

*13. Write a program to find the number of student users in the college, get the total users, staff users details from the client. Note for every 3 staff user there is one Non teaching staff user assigned by default.*

**Program:**

total\_users = int(input("Enter the total number of users: "))

staff\_users = int(input("Enter the number of staff users: "))

non\_teaching\_staff\_users = staff\_users // 3

student\_users = total\_users - staff\_users - non\_teaching\_staff\_users

print(f"Student users: {student\_users}")

**Sample Input:**

Total Users: 856

Staff Users: 126

**Sample Output:**

Student Users: 688

*14. Write the string in the upper case and count the number of spaces available in the string.*

**Program:**

a = "Python is the interpreted language"

a\_upper = a.upper()

num\_spaces = a.count(" ")

print("Uppercase string:", a\_upper)

print("Number of spaces:", num\_spaces)

*15. Merge 2 lists:*

**Program:**

a = [1, 2, 3, 4, 5]

b = [11, 12, 13]

c = a + b

print(c)

**input:**

a= [1,2,3,4,5]

b= [11,12,13]

**Output:**

c= [1,2,3,4,5,11,12,13]

*16. FInd the number of words in the sentence:*

**Program:**

sentence = input("Enter a sentence: ")

words = sentence.split()

num\_words = len(words)

print("Number of words in the sentence:", num\_words)

**Input:** Python is a high level interpreted programming language

**Output:** 9

*17: Convert the every word first letter to upper case letter and join (.).*

**Program:**

sentence = input("Enter a sentence: ")

words = sentence.split()

modified\_words = []

for word in words:

modified\_word = word[0].upper() + word[1:]

modified\_words.append(modified\_word)

result = '.'.join(modified\_words)

print("Output:", result)

**Input:** this is  a cat

**Output:** T.I.A.C

*18. Print the given number is prime or not*

**Program:**

def is\_prime(n):

"""Returns True if the given number is prime, False otherwise."""

if n <= 1:

return False

for i in range(2, int(n \*\* 0.5) + 1):

if n % i == 0:

return False

return True

num = int(input("Enter a number: "))

result = is\_prime(num)

if result:

print(f"{num} is a prime number")

else:

print(f"{num} is not a prime number")

**Input:** 12

**Output:** False

*19. Find the maximum of three binary values using looping*

**Program:**

def binary\_to\_decimal(binary\_str):

decimal\_val = 0

for i in range(len(binary\_str)):

if binary\_str[i] == '1':

decimal\_val += 2 \*\* (len(binary\_str) - i - 1)

return decimal\_val

binary1 = input("Enter binary value 1: ")

binary2 = input("Enter binary value 2: ")

binary3 = input("Enter binary value 3: ")

decimal1 = binary\_to\_decimal(binary1)

decimal2 = binary\_to\_decimal(binary2)

decimal3 = binary\_to\_decimal(binary3)

max\_decimal = max(decimal1, decimal2, decimal3)

max\_binary = bin(max\_decimal).replace("0b", "")

print("Maximum binary value:", max\_binary)

**Sample Input:**

Given Numbers: 1101, 1011, 1001

**Sample Output:**

Maximum Number: 1101

*20: Write a program that accepts a string from user and re displays the same string after removing vowels from it.*

**Program:**

input\_string = input("Enter a string: ")

def remove\_vowels(s):

vowels = "AEIOUaeiou"

return "".join(char for char in s if char not in vowels)

output\_string = remove\_vowels(input\_string)

print("After removing vowels:", output\_string)

**Sample Input & Output:**

Enter a string: we can play the game

The string without vowels is: w cn ply th gm