**1.create a pyramid of numbers from 1 to 20 using for loop?**

def print\_pyramid(n):

current\_num = 1

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

for \_ in range(n-i):

print(" ", end=" ")

for \_ in range(i):

if current\_num > 20:

break

print(current\_num, end=" ")

current\_num += 1

print() # New line after each row

print\_pyramid(20)

**2. Python program that calculates the total number of vowels and the count of each individual vowel (A, E, I, O, U) in the given string "Guvi Geeks Network Private Limited":**

def count\_vowels(s):

s = s.upper()

vowel\_counts = {'A': 0, 'E': 0, 'I': 0, 'O': 0, 'U': 0}

for char in s:

if char in vowel\_counts:

vowel\_counts[char] += 1

total\_vowels = sum(vowel\_counts.values())

return total\_vowels, vowel\_counts

input\_string = "Guvi Geeks Network Private Limited"

total\_vowels, vowel\_counts = count\_vowels(input\_string)

print(f"Total number of vowels: {total\_vowels}")

print("Count of each vowel:")

for vowel, count in vowel\_counts.items():

print(f"{vowel}: {count}")

**3.write a string that takes the string and returns a new string with all the vowels removed**

def remove\_vowels(s):

vowels = "AEIOUaeiou"

result = ""

for char in s:

if char not in vowels:

result += char

return result

input\_string = "Guvi Geeks Network Private Limited"

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

print("Original string:", input\_string)

print("String with vowels removed:", output\_string)

**4.write a program that takes a string and returns the number of unique characters in it .**

def count\_unique\_characters(s):

unique\_chars = set(s)

return len(unique\_chars)

input\_string = "Guvi Geeks Network Private Limited"

unique\_char\_count = count\_unique\_characters(input\_string)

print("Original string:", input\_string)

print("Number of unique characters:", unique\_char\_count)

**5.write a program that takes a string and returns true if it is a palindrome,false otherwise**

def is\_palindrome(s):

cleaned\_string = ''.join(char.lower() for char in s if char.isalnum())

return cleaned\_string == cleaned\_string[::-1]

test\_strings = [

"A man, a plan, a canal, Panama",

"racecar",

"hello",

"No 'x' in Nixon",

"Was it a car or a cat I saw?"

]

for test in test\_strings:

print(f"'{test}' -> {is\_palindrome(test)}")

**6.Certainly! Here's a Python program that takes two strings as input and returns the longest common substring between them:**

def longest\_common\_substring(str1, str2):

m = len(str1)

n = len(str2)

# Create a 2D list to store lengths of longest common suffixes of substrings

# dp[i][j] will store the length of longest common substring ending with str1[i-1] and str2[j-1]

dp = [[0] \* (n + 1) for \_ in range(m + 1)]

max\_length = 0

end\_index = 0

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

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

if str1[i - 1] == str2[j - 1]:

dp[i][j] = dp[i - 1][j - 1] + 1

if dp[i][j] > max\_length:

max\_length = dp[i][j]

end\_index = i

else:

dp[i][j] = 0

return longest\_substring

# Example usage:

str1 = "abcdefg"

str2 = "xyabcfghi"

print("Longest common substring:", longest\_common\_substring(str1, str2))

**7.write a program that takes a string as input and returns the most frequent character in it:**

def most\_frequent\_character(input\_string):

char\_frequency = {}

for char in input\_string:

if char in char\_frequency:

char\_frequency[char] += 1

else:

char\_frequency[char] = 1

max\_frequency = 0

most\_frequent\_char = ''

for char, freq in char\_frequency.items():

if freq > max\_frequency:

max\_frequency = freq

most\_frequent\_char = char

return most\_frequent\_char

# Example usage:

input\_string = "hello world"

print("Most frequent character:", most\_frequent\_character(input\_string))

**8. write a program that takes a string and returns true if it is an anagram of another string false otherwise**

def are\_anagrams(str1, str2):

# Remove spaces and convert both strings to lowercase

str1 = str1.replace(" ", "").lower()

str2 = str2.replace(" ", "").lower()

if len(str1) != len(str2):

return False

sorted\_str1 = sorted(str1)

sorted\_str2 = sorted(str2)

if sorted\_str1 == sorted\_str2:

return True

else:

return False

string1 = "listen"

string2 = "silent"

print("Are they anagrams?", are\_anagrams(string1, string2))

**9.write a program that takes a string and returns the number of words in it**

def count\_words(input\_string):

words\_list = input\_string.split()

return len(words\_list)

# Example usage:

input\_string = "Hello, this is a sample string."

print("Number of words:", count\_words(input\_string))