**COURSE CODE : CSA0836**

**COURSE : PYTHON PROGRAMMING**

**NAME : N.Bala Adithya**

**REGISTER NUMBER : 192111151**

**TEST – 4**

Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2.

You have the following three operations permitted on a word: Insert a character Delete a character Replace a character

Test case: 1.Input: word1 = "horse", word2 = "ros" Output: 3 2.Input: word1 = "intention", word2 = "execution" Output: 5 3.Input: str1 = “sunday”, str2 = “saturday” Output: 3 4.Input: str1 = “cat”, str2 = “cut” Output: 1 5.Input: str1 = “girl”, str2 = “grill” Output: 2

**def wordcovert(str1, str2, m, n):**

**if m == 0:**

**return n**

**if n == 0:**

**return m**

**if str1[m-1] == str2[n-1]:**

**return wordcovert(str1, str2, m-1, n-1)**

**return 1 + min(wordcovert(str1, str2, m, n-1),**

**wordcovert(str1, str2, m-1, n),**

**wordcovert(str1, str2, m-1, n-1)**

**)**

**str1 =input("Enter Your String1 : ")**

**str2 = input("Enter Your String : ")**

**print (wordcovert(str1, str2, len(str1), len(str2)))**

2Q Add Binary Given two binary strings a and b, return their sum as a binary string.

a and b consist only of '0' or '1' characters.

Each string does not contain leading zeros except for the zero itself.

Test cases: 1.Input: a = "11", b = "1" Output: "100" 2.Input: a = "1010", b = "1011" Output: "10101" 3.a= “1111”, b= “1010” 4.a= “101101”, b= “1100” 5.a= “15” b= “45”

**a=input("Enter first binary number:")**

**b=input("Enter second number:")**

**sum=bin(int(a,2)+int(b,2))**

**print("The sum of two binary numbers is:",sum[2:])**

3Q Permutations Given a collection of numbers, nums, that might contain duplicates, return all possible unique permutations in any order.

**from itertools import permutations**

**a=eval(input("Enter list:"))**

**lst=permutations(a)**

**for i in list(lst):**

**print(i)**