**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе

«Расстояние Левенштейна»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Абрамов А. В. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

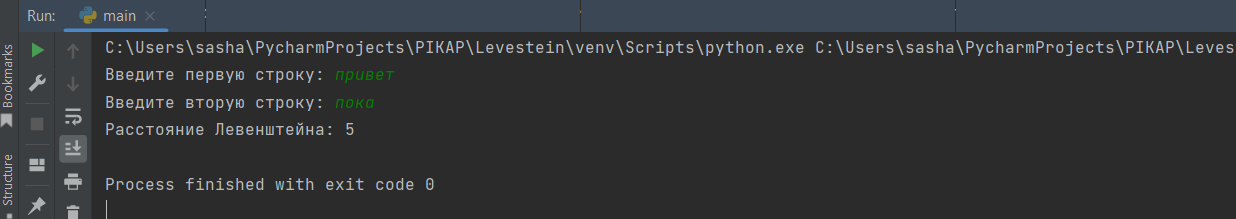
**Описание задания**

Разработать программу для нахождения расстояния Левенштейна между двумя словами.

**Текст программы Python**

def levenshtein\_distance(str1, str2):  
 len1, len2 = len(str1), len(str2)  
 dp = [[0] \* (len2 + 1) for \_ in range(len1 + 1)]  
  
 for i in range(len1 + 1):  
 dp[i][0] = i  
 for j in range(len2 + 1):  
 dp[0][j] = j  
  
 for i in range(1, len1 + 1):  
 for j in range(1, len2 + 1):  
 if str1[i - 1] == str2[j - 1]:  
 cost = 0  
 else:  
 cost = 1  
  
 dp[i][j] = min(  
 dp[i - 1][j] + 1, # Удаление  
 dp[i][j - 1] + 1, # Вставка  
 dp[i - 1][j - 1] + cost, # Замена  
 )  
  
 return dp[len1][len2]  
  
  
def main():  
 print("Введите первую строку: ", end='')  
 a = input()  
 print("Введите вторую строку: ", end='')  
 b = input()  
  
 print("Расстояние Левенштейна: {}".format(levenshtein\_distance(a, b)))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Примеры выполнения программы**

****

**Тест программы C#**

using System;

namespace ConsoleApp1

{

class Program

{

static int levenshtein\_distance(string str1, string str2)

{

int len1 = str1.Length;

int len2 = str2.Length;

int[,] dp = new int[len1 + 1, len2 + 1];

for (int i = 0; i <= len1; i++)

dp[i, 0] = i;

for (int j = 0; j <= len2; j++)

dp[0, j] = j;

for (int i = 1; i <= len1; i++)

{

for (int j = 1; j <= len2; j++)

{

int cost = (str1[i - 1] == str2[j - 1]) ? 0 : 1;

dp[i, j] = Math.Min(

dp[i - 1, j] + 1,

Math.Min(

dp[i, j - 1] + 1,

dp[i - 1, j - 1] + cost

)

);

}

}

return dp[len1, len2];

}

static void Main(string[] args)

{

Console.Write("Введите первую строку: ");

string str1 = Console.ReadLine();

Console.Write("Введите вторуб строку: ");

string str2 = Console.ReadLine();

int distance = levenshtein\_distance(str1, str2);

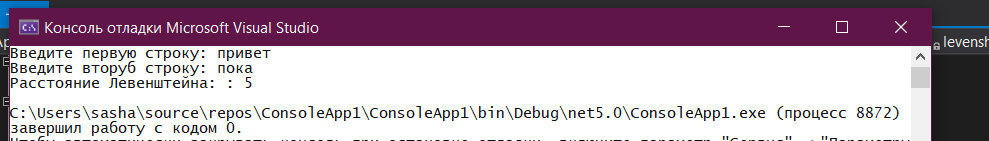
Console.WriteLine($"Расстояние Левенштейна: : {distance}");

}

}

}

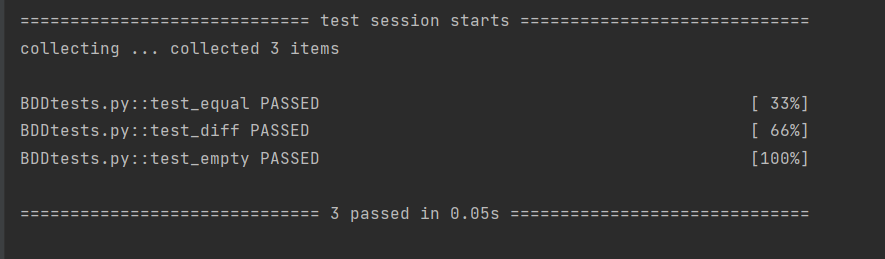
**Пример выполнения программы**



**Текст программы тестов на BDD**

from main import levenshtein\_distance  
import pytest  
  
  
def test\_equal():  
 assert levenshtein\_distance("Валера", "Валера") == 0  
  
  
def test\_diff():  
 assert levenshtein\_distance("Разведка", "Разводка") == 1  
  
  
def test\_empty():  
 assert levenshtein\_distance("", "Привет") == 6  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 pytest.main()

**Пример прохождения тестов**



**Текст программы тестов на TDD**

import unittest  
from main import levenshtein\_distance  
  
  
class TestLevenshteinDistance(unittest.TestCase):  
 def test\_equal(self):  
 self.assertEqual(levenshtein\_distance("Валера", "Валера"), 0)  
  
 def test\_diff(self):  
 self.assertEqual(levenshtein\_distance("Разведка", "Разводка"), 1)  
  
 def test\_empty(self):  
 self.assertEqual(levenshtein\_distance("", "Привет"), 6)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Пример прохождения тестов**

