

**Московский государственный технический
университет им. Н. Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Система обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»
Лабораторная работа на тему “Расстояние левенштейна”

Выполнил:

Студент ИУ5-34Б

Фролов М. К.

Подпись и Дата:

Проверил:

Преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и Дата:

Москва, 2023 г

Задание:

Разработать программы, которая будет высчитывать расстояние Левенштейна между двумя словами.

Реализация:

```
# Алгоритм Вагнера-Фишера
#           e d i t i n g
#           0 1 2 3 4 5 6 7
#
# 0 | 0 1 2 3 4 5 6 7
# d 1 | 1 1 1 2 3 4 5 6
# i 2 | 2 2 2 1 2 3 4 5
# s 3 | 3 3 3 2 2 3 4 5
# t 4 | 4 4 4 3 2 3 4 5
# a 5 | 5 5 5 4 3 3 4 5
# n 6 | 6 6 6 5 4 4 3 4
# c 7 | 7 7 7 6 5 5 4 4
# e 8 | 8 7 8 7 6 6 5 5
def Levenshtein_distance(s1, s2):
    m = len(s1)
    n = len(s2)
    matrix = []
    for i in range(n + 1): matrix.append([0] * (m + 1))
    for i in range(m + 1): matrix[0][i] = i
    for i in range(n + 1): matrix[i][0] = i
    for i in range(1, n + 1):
        for j in range(1, m + 1):
            if s1[j - 1] != s2[i - 1]:
                matrix[i][j] = min(matrix[i - 1][j] + 1, matrix[i - 1][j - 1] +
1, matrix[i][j - 1] + 1)
            else:
                matrix[i][j] = min(matrix[i - 1][j] + 1, matrix[i - 1][j - 1] +
0, matrix[i][j - 1] + 1)
    return matrix[n][m]
print("\nВВЕДИТЕ ДВЕ СТРОКИ:\n")
s1 = input()
s2 = input()
print(Levenshtein_distance(s1, s2))
```

Результат работы программы:

```
биба
боба
1
PS C:\Users\fmfmf2>
```