



Московский государственный технический университет

им. Н.Э. Баумана

(МГТУ им. Н.Э. Баумана)

Кафедра «Системы обработки информации и управления» (ИУ5)

Лабораторная работа № 2

По дисциплине: «Парадигмы и конструкции языков программирования»

**«ПРОГРАММА ДЛЯ ПОИСКА РАССТОЯНИЯ ДАМЕРАУ-
ЛЕВЕНШТЕЙНА»**

Выполнила: Вешторт Е. С.,

студентка группы ИУ5-32Б

Проверил: Нардид А. Н.

г. Москва 2023 г.

Задание: написать программу для поиска расстояния Дameraу-Левенштейна между строками.

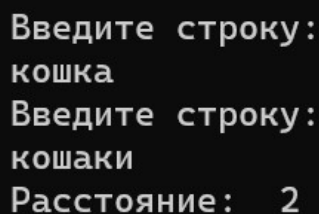
```
import sys

def get_str(index):
    try:
        a = sys.argv[index]
    except:
        print("Введите строку:")
        a = input()
    return a

def damerau_levenshtein(a, b):
    n = len(a)
    m = len(b)
    matrix = []
    for i in range(m+1):
        matrix.append([0] * (n+1))
        for j in range(n+1):
            if i == 0:
                matrix[i][j] = j
            elif j == 0:
                matrix[i][j] = i
            elif i == 1 or j == 1:
                if a[j-1] == b[i-1]: k = 0
                else: k = 1
                matrix[i][j] = min(matrix[i-1][j] + 1, matrix[i][j-1] + 1, matrix[i-1][j-1] + k)
            else:
                if a[j-1] == b[i-1]: k = 0
                else: k = 1
                if a[j-2] == b[i-1] and a[j-1] == b[i-2]:
                    matrix[i][j] = min(matrix[i-1][j] + 1, matrix[i][j-1] + 1, matrix[i-1][j-1] + k, matrix[i-2][j-2] + k)
                else:
                    matrix[i][j] = min(matrix[i-1][j] + 1, matrix[i][j-1] + 1, matrix[i-1][j-1] + k)
    return(matrix[m][n])

def main():
    str1 = get_str(1)
    str2 = get_str(2)
    print("Расстояние: ", damerau_levenshtein(str1, str2))

if __name__ == "__main__":
    main()
```



```
Введите строку:
кошка
Введите строку:
кошаки
Расстояние: 2
```