МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южно-Уральский государственный университет

(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

**ОТЧЕТ**

о выполнении практического задания №1

по дисциплине

«Структуры и алгоритмы обработки данных»

|  |  |
| --- | --- |
|  | Выполнил:  студент группы КЭ-142  Замятин Андрей Александрович  Проверил:  ст. преподаватель кафедры СП  Валиулин А.А. |

Челябинск – 2024

**Словесная постановка задачи:**

Цель данной практической работы заключается в создании программы, выполняющей нормирование введенной строки. Вводится ненормированная строка, где могут присутствовать пробелы в начале, в конце и между словами более одного пробела. Требуется привести ее к нормированному виду, удалив все пробелы в начале и конце, а между словами оставив только один пробел.

**Листинг программы:**

not\_normalized\_string = input("Введите ненормированную строку: ")

# Удаляем все пробелы из строки

normalized\_string = not\_normalized\_string.strip().replace("  ", " ")

# Выводим нормированную строку

print("Нормированная строка:", normalized\_string)

**Контрольный тест:**

Входные данные:

Введите ненормированную строку: Пример не нормированной строки.

Результат выполнения программы:

Нормированная строка: Пример не нормированной строки.

**Выводы по практической работе:**

В ходе выполнения практической работы была разработана программа на языке программирования Python, которая успешно выполняет задачу нормирования введенной строки. Программа удаляет лишние пробелы в начале и конце строки, а также заменяет последовательности пробелов между словами на одиночные пробелы. Таким образом, поставленная задача была выполнена успешно, что делает программу полезным инструментом для обработки строковых данных.

**Словесная постановка задачи:**

Цель данной практической работы заключается в создании программы, которая находит сумму отрицательных элементов одномерного целочисленного массива, расположенных после последнего положительного элемента. Если в массиве нет положительных элементов, программа должна вернуть значение 0.

**Листинг программы:**

import random

n = int(input("Введите количество элементов массива: "))

pos\_num = 0

last\_pos\_num\_id = -1

sum\_neg\_after\_pos = 0

# Задаём массив из n элементов целых случайных чисел в диапазоне от -1000 до 1000

random\_massif = [random.randint(-1000, 1000) for i in range(n)]

print(random\_massif)

# Перебираем массив для поиска последнего положительного элемента

for i in range(n):

if random\_massif[i] > 0:

pos\_num = random\_massif[i]

last\_pos\_num\_id = i

# Суммируем все отрицательные элементы после последнего положительного элемента

if last\_pos\_num\_id != -1:

for j in range(last\_pos\_num\_id + 1, n):

if random\_massif[j] < 0:

sum\_neg\_after\_pos += random\_massif[j]

print("Сумма отрицательных элементов, стоящих после последнего положительного элемента: ", sum\_neg\_after\_pos)

**Контрольный тест:**

*Входные данные:*

Введите количество элементов массива: 10

Результат выполнения программы:

[-543, -401, 619, 635, 913, 942, 111, 576, -692, -348]

Сумма отрицательных элементов, стоящих после последнего положительного элемента: -1040

**Выводы по практической работе:**

В ходе выполнения практической работы была разработана программа на языке программирования Python, которая успешно находит сумму отрицательных элементов массива, расположенных после последнего положительного элемента. Программа создает массив случайных целых чисел, определяет последний положительный элемент, а затем находит и суммирует отрицательные элементы, следующие за ним. Таким образом, поставленная задача была выполнена успешно.