### **Лабораторная работа №7**

**Тема: Работа со строками**

### **Задача 1. Условие.**

Программа для подсчета числа слов во введенной с клавиатуры строке. Словом считается любая последовательность символов, разделенная пробелами.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | word\_count | Количество слов | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Привет, как дела?" |
| Выходные данные | word\_count | 3 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
word\_count = len(input\_string.split())  
print(f"Количество слов: {word\_count}")

### **Задача 2. Условие.**

Программа для проверки, является ли введенная строка числом в двоичной системе.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | binary\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | is\_binary | Логическое значение (да/нет) | логическое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | binary\_string | "101010" |
| Выходные данные | is\_binary | True |

**Листинг программы:**

binary\_string = input("Введите строку: ")  
is\_binary = all(c in "01" for c in binary\_string)  
if is\_binary:  
print("Строка является числом в двоичной системе")  
else:  
print("Строка не является числом в двоичной системе")

### **Задача 3. Условие.**

Программа для вычисления средней длины слов во введенной строке.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | words | Список слов из строки | список строк |  |
| Выходные данные | average\_length | Средняя длина слов | вещественное |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Это пример строки" |
| Выходные данные | average\_length | 4.33 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
average\_length = sum(len(word) for word in words) / len(words) if words else 0  
print(f"Средняя длина слов: {average\_length:.2f}")

### **Задача 4. Условие.**

Программа для шифрования введенного текста путем изменения кода символов.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_text | Текст, введенный пользователем | строка |  |
| Выходные данные | encrypted\_text | Зашифрованный текст | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_text | "Hello" |
| Выходные данные | encrypted\_text | "&>>aa" |

**Листинг программы:**

input\_text = input("Введите текст для шифрования: ")  
encrypted\_text = "".join(chr(ord(c) - 10) for c in input\_text)  
print(f"Зашифрованный текст: {encrypted\_text}")

### **Задача 5. Условие.**

Программа для определения, является ли слово палиндромом.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | word | Слово, введенное пользователем | строка |  |
| Выходные данные | is\_palindrome | Логическое значение (да/нет) | логическое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | word | "потоп" |
| Выходные данные | is\_palindrome | True |

**Листинг программы:**

word = input("Введите слово: ")  
is\_palindrome = word == word[::-1]  
print("Слово является палиндромом" if is\_palindrome else "Слово не является палиндромом")

### **Задача 6. Условие.**

Программа для нахождения самого длинного слова в строке и количества слов такой же длины.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | max\_length | Длина самого длинного слова | целое |  |
| Промежуточные данные | count\_max\_length | Количество слов такой же длины | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Один два три четыре пять" |
| Выходные данные | max\_length | 7 |
| Выходные данные | count\_max\_length | 1 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
if words:  
max\_length = max(len(word) for word in words)  
count\_max\_length = sum(1 for word in words if len(word) == max\_length)  
print(f"Самое длинное слово имеет длину {max\_length} и встречается {count\_max\_length} раз(а).")  
else:  
print("Строка пуста")

### **Задача 7. Условие.**

Программа для удаления всех пробелов из строки.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | result | Строка без пробелов | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Это пример строки" |
| Выходные данные | result | "Этопримерстроки" |

**Листинг программы:**

input\_string = input("Введите строку: ")  
result = input\_string.replace(" ", "")  
print(f"Строка без пробелов: {result}")

### **Задача 8. Условие.**

Программа для удаления указанного слова из строки.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Входные данные | word\_to\_remove | Слово для удаления | строка |  |
| Выходные данные | result | Строка после удаления слова | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Это пример строки для удаления слова" |
| Входные данные | word\_to\_remove | "пример" |
| Выходные данные | result | "Это строки для удаления слова" |

**Листинг программы:**

input\_string = input("Введите строку: ")  
word\_to\_remove = input("Введите слово для удаления: ")  
result = input\_string.replace(word\_to\_remove, "")  
print(f"Строка после удаления слова: {result}")

### **Задача 9. Условие.**

Программа для выделения подстроки между первой и второй точками в строке.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | first\_dot | Индекс первой точки в строке | целое |  |
| Промежуточные данные | second\_dot | Индекс второй точки в строке | целое |  |
| Выходные данные | substring | Подстрока между первой и второй точками | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Это. пример. строки." |
| Выходные данные | substring | " пример" |

**Листинг программы:**

input\_string = input("Введите строку: ")  
first\_dot = input\_string.find(".")  
second\_dot = input\_string.find(".", first\_dot + 1)  
if first\_dot != -1 and second\_dot != -1:  
substring = input\_string[first\_dot + 1

]  
print(f"Подстрока между первой и второй точками: {substring}")  
else:  
print("Недостаточно точек в строке")

### **Задача 10. Условие.**

Программа для определения длины самого короткого и самого длинного слова в строке.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | min\_length | Длина самого короткого слова | целое |  |
| Промежуточные данные | max\_length | Длина самого длинного слова | целое |  |
| Выходные данные | result | Длины самого короткого и длинного слова | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Это пример для теста" |
| Выходные данные | min\_length | 3 |
| Выходные данные | max\_length | 6 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
if words:  
min\_length = len(min(words, key=len))  
max\_length = len(max(words, key=len))  
print(f"Длина самого короткого слова: {min\_length}, длина самого длинного слова: {max\_length}")  
else:  
print("Строка пуста")

### **Задача 11. Условие.**

Программа для определения количества слов, начинающихся и заканчивающихся одной и той же буквой.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | count | Количество слов, начинающихся и заканчивающихся одной буквой | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Анна видит диван" |
| Выходные данные | count | 1 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
count = sum(1 for word in words if word[0].lower() == word[-1].lower())  
print(f"Количество слов, начинающихся и заканчивающихся одной и той же буквой: {count}")

### **Задача 12. Условие.**

Программа для определения количества слов, содержащих хотя бы одну букву "е".

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | count | Количество слов, содержащих букву "е" | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Семь елок возле ели" |
| Выходные данные | count | 3 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
count = sum(1 for word in words if "е" in word.lower())  
print(f"Количество слов, содержащих хотя бы одну букву 'е': {count}")

### **Задача 13. Условие.**

Программа для проверки, является ли строка правильным скобочным выражением (с учетом только круглых скобок).

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | balance | Текущий баланс скобок | целое |  |
| Выходные данные | is\_correct | Логическое значение правильности | логическое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "(a+b)\*(c-d)" |
| Выходные данные | is\_correct | True |

**Листинг программы:**

input\_string = input("Введите строку: ")  
balance = 0  
for char in input\_string:  
if char == "(":  
balance += 1  
elif char == ")":  
balance -= 1  
if balance < 0:  
break  
if balance == 0:  
print("Строка является правильным скобочным выражением")  
else:  
print("Строка не является правильным скобочным выражением")

### **Задача 14. Условие.**

Программа для определения количества слов, содержащих ровно три буквы "е".

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | count | Количество слов с ровно тремя буквами "е" | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "зеленеет зелеенееее зелёные" |
| Выходные данные | count | 2 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
count = sum(1 for word in words if word.lower().count("е") == 3)  
print(f"Количество слов, содержащих ровно три буквы 'е': {count}")

### **Задача 15. Условие.**

Программа для удаления всех впереди стоящих нулей из строки.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, содержащая только цифры | строка |  |
| Выходные данные | result | Строка после удаления всех нулей | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "0001234" |
| Выходные данные | result | "1234" |

**Листинг программы:**

input\_string = input("Введите строку, содержащую только цифры: ")  
result = input\_string.lstrip("0")  
print(f"Строка после удаления всех впереди стоящих нулей: {result}")

### **Задача 16. Условие.**

Программа для подсчета количества знаков препинания в строке.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | punctuation\_count | Количество знаков препинания | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Привет! Как дела? Все хорошо." |
| Выходные данные | punctuation\_count | 3 |

**Листинг программы:**

import string  
input\_string = input("Введите строку: ")  
punctuation\_count = sum(1 for char in input\_string if char in string.punctuation)  
print(f"Количество знаков препинания в строке: {punctuation\_count}")

### **Задача 17. Условие.**

Программа для удаления всех запятых из строки.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | result | Строка после удаления всех запятых | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Привет, как дела, все хорошо?" |
| Выходные данные | result | "Привет как дела все хорошо?" |

**Листинг программы:**

input\_string = input("Введите строку: ")  
result = input\_string.replace(",", "")  
print(f"Строка после удаления всех запятых: {result}")

### **Задача 18. Условие.**

Программа для вставки слова после каждого пробела в строке.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Входные данные | word\_to\_insert | Слово для вставки | строка |  |
| Выходные данные | result | Строка после вставки слова | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Это пример текста" |
| Входные данные | word\_to\_insert | "вставка" |
| Выходные данные | result | "Это вставка пример вставка текста" |

**Листинг программы:**

input\_string = input("Введите строку: ")  
word\_to\_insert = input("Введите слово для вставки: ")  
result = input\_string.replace(" ", " " + word\_to\_insert + " ")  
print(f"Строка после вставки слова: {result}")

### **Задача 19. Условие.**

Программа для нахождения суммы чисел, встречающихся в строке.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | numbers | Числа, найденные в строке | список целых |  |
| Выходные данные | sum\_of\_numbers | Сумма чисел в строке | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Встретились числа 5 и 15, а также 3" |
| Выходные данные | sum\_of\_numbers | 23 |

**Листинг программы:**

import re  
input\_string = input("Введите строку: ")  
numbers = map(int, re.findall(r"\d+", input\_string))  
sum\_of\_numbers = sum(numbers)  
print(f"Сумма чисел в строке: {sum\_of\_numbers}")

### **Задача 20. Условие.**

Программа для удаления всех чисел из строки.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | result | Строка после удаления всех чисел | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Встретились числа 5 и 15, а также 3" |
| Выходные данные | result | "Встретились числа и , а также " |

**Листинг программы:**

import re  
input\_string = input("Введите строку: ")  
result = re.sub(r"\d+", "", input\_string)  
print(f"Строка после удаления всех чисел: {result}")

### **Задача 21. Условие.**

Программа для удаления слова с наибольшей длиной из строки.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Промежуточные данные | longest\_word | Самое длинное слово в строке | строка |  |
| Выходные данные | result | Строка после удаления самого длинного слова | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Этот текст содержит длинное слово" |
| Выходные данные | result | "Этот текст содержит слово" |

**Листинг программы:**

input\_string = input("Введите строку: ")  
words = input\_string.split()  
if words:  
longest\_word = max(words, key=len)  
result = input\_string.replace(longest\_word, "", 1)  
print(f"Строка после удаления самого длинного слова: {result.strip()}")  
else:  
print("Строка пуста")

### **Задача 22. Условие.**

Программа для вставки пробела после каждого знака препинания.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | result | Строка после вставки пробелов | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Привет!Как дела?" |
| Выходные данные | result | "Привет! Как дела? " |

**Листинг программы:**

import re  
input\_string = input("Введите строку: ")  
result = re.sub(r"([.,!?;:])", r"\1 ", input\_string)  
print(f"Строка после вставки пробелов: {result}")

### **Задача 23. Условие.**

Программа для определения, каких букв больше в строке: гласных или согласных.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | vowels\_count | Количество гласных букв | целое |  |
| Выходные данные | consonants\_count | Количество согласных букв | целое |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | "Hello, world!" |
| Выходные данные | vowels\_count | 3 |
| Выходные данные | consonants\_count | 7 |

**Листинг программы:**

input\_string = input("Введите строку: ")  
vowels = "AEIOUaeiou"  
consonants\_count = sum(1 for char in input\_string if char.isalpha() and char not in vowels)  
vowels\_count = sum(1 for char in input\_string if char in vowels)  
if vowels\_count > consonants\_count:  
print("Гласных больше")  
else:  
print("Согласных больше")

### **Задача 24. Условие.**

Программа для удаления частей строки, заключенных в кавычки (вместе с кавычками).

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | input\_string | Строка, введенная пользователем | строка |  |
| Выходные данные | result | Строка после удаления частей в кавычках | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | input\_string | 'Это "тест" строки "для удаления"' |
| Выходные данные | result | 'Это строки ' |

**Листинг программы:**

import re  
input\_string = input("Введите строку: ")  
result = re.sub(r"".\*?"", "", input\_string)  
print(f"Строка после удаления частей в кавычках: {result.strip()}")

### **Задача 25. Условие.**

Программа для замены буквы "A" на букву "O" в слове. Если буквы "A" нет, выводится сообщение.

**Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | word | Слово, введенное пользователем | строка |  |
| Выходные данные | result | Слово после замены | строка |  |

**Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | word | "Карта" |
| Выходные данные | result | "Корто" |

**Листинг программы:**

word = input("Введите слово: ")  
if "A" in word.upper():  
result = word.replace("A", "O").replace("a", "o")  
print(f"Результат замены: {result}")  
else:  
print("Буквы A в слове нет")