Лабараторная работа №10

### **Задача 1. Условие.**

Подсчитайте количество сдвоенных символов 'аа', 'оо', 'кк' в тексте, расположенном в текстовом файле, затем удалите повторяющийся символ. Полученную строку запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие букв |
| Выходные данные | modified\_text | Текст после удаления повторяющихся символов | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "коооорова кааак осооообенно ккрасиво выглядит" |
| Выходные данные | modified\_text | "корова как особенно красиво выглядит" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

double\_chars = ['аа', 'оо', 'кк']  
count = sum(text.count(pair) for pair in double\_chars)  
print(f"Количество сдвоенных символов: {count}")

modified\_text = text.replace('аа', 'а').replace('оо', 'о').replace('кк', 'к')

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(modified\_text)

### **Задача 2. Условие.**

Подсчитайте число слов в предложении, записанном в текстовом файле.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | sentence | Предложение из файла | строка | наличие слов |
| Выходные данные | word\_count | Количество слов | целое число |  |

Тесты.

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

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
sentence = file.read()

word\_count = len(sentence.split())  
print(f"Количество слов: {word\_count}")

### **Задача 3. Условие.**

Найдите в текстовом файле самое длинное и самое короткое слово.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | words | Список слов | список строк | наличие слов |
| Выходные данные | longest\_word | Самое длинное слово | строка |  |
| Выходные данные | shortest\_word | Самое короткое слово | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | words | "Это тестовая строка для задачи" |
| Выходные данные | longest\_word | "тестовая" |
| Выходные данные | shortest\_word | "Это" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
words = file.read().split()

longest\_word = max(words, key=len)  
shortest\_word = min(words, key=len)

print(f"Самое длинное слово: {longest\_word}")  
print(f"Самое короткое слово: {shortest\_word}")

### **Задача 4. Условие.**

Из строки, расположенной в текстовом файле, исключите все символы, входящие в нее более одного раза. Полученную строку сохраните в другом файле.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Строка из файла | строка | наличие символов |
| Выходные данные | modified\_text | Строка без повторяющихся символов | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "абракадабра" |
| Выходные данные | modified\_text | "кд" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

modified\_text = ''.join([char for char in text if text.count(char) == 1])

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(modified\_text)

### **Задача 5. Условие.**

Проверьте, правильно ли расставлены в тексте, расположенном в текстовом файле, круглые скобки. Если неправильно — исправить, а результат сохранить в файле.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст с круглыми скобками | строка | наличие скобок |
| Выходные данные | is\_correct | Логическое значение правильности | логическое |  |

Тесты.

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

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

balance = 0  
for char in text:  
if char == "(":  
balance += 1  
elif char == ")":  
balance -= 1  
if balance < 0:  
break

is\_correct = (balance == 0)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write("Скобки расставлены правильно" if is\_correct else "Скобки расставлены неправильно")

### **Задача 6. Условие.**

В последовательности символов, заданной в текстовом файле, подсчитайте общее количество символов '+', '-', '\*'.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Строка символов | строка | наличие символов |
| Выходные данные | count | Количество символов '+', '-', '\*' | целое |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "a + b - c \* d" |
| Выходные данные | count | 3 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

count = sum(text.count(char) for char in '+-*')  
print(f"Количество символов '+', '-', '*': {count}")

### **Задача 7. Условие.**

В текстовом файле, в предложении, содержащем не менее двух слов, поменяйте местами первое и последнее слова, а затем результат сохраните в другом файле.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | sentence | Предложение из файла | строка | наличие слов |
| Выходные данные | modified\_sentence | Измененное предложение | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | sentence | "Привет как дела" |
| Выходные данные | modified\_sentence | "дела как Привет" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
sentence = file.read().strip()

words = sentence.split()  
if len(words) >= 2:  
words[0], words[-1] = words[-1], words[0]  
modified\_sentence = ' '.join(words)  
else:  
modified\_sentence = sentence

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(modified\_sentence)

### **Задача 8. Условие.**

В текстовом файле две строки текста. Необходимо сформировать третью строку, состоящую из символов, входящих одновременно в обе исходные строки, и дописать ее в текстовый файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | str1 | Первая строка | строка | наличие символов |
| Входные данные | str2 | Вторая строка | строка | наличие символов |
| Выходные данные | common\_chars | Общие символы | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | str1 | "abcdef" |
| Входные данные | str2 | "defghi" |
| Выходные данные | common\_chars | "def" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()  
str1, str2 = lines[0].strip(), lines[1].strip()

common\_chars = ''.join(set(str1).intersection(str2))

with open('output.txt', 'a', encoding='utf-8') as file:  
file.write(f"Общие символы: {common\_chars}\n")

### **Задача 9. Условие.**

Откорректируйте текст, расположенный в текстовом файле, заменив в нем все вхождения одной буквы на другую. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие символов |
| Входные данные | old\_char | Заменяемая буква | строка | одиночный символ |
| Входные данные | new\_char | Новая буква | строка | одиночный символ |
| Выходные данные | modified\_text | Текст с замененными буквами | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "Привет мир" |
| Входные данные | old\_char | "и" |
| Входные данные | new\_char | "е" |
| Выходные данные | modified\_text | "Превет мер" |

Листинг программы:  
old\_char = input("Введите заменяемую букву: ")  
new\_char = input("Введите новую букву: ")

with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

modified\_text = text.replace(old\_char, new\_char)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(modified\_text)

### **Задача 10. Условие.**

Перепишите текстовый файл таким образом, чтобы все слова исходного текста были перевернуты. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие слов |
| Выходные данные | reversed\_text | Текст с перевернутыми словами | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "Привет мир" |
| Выходные данные | reversed\_text | "тевирП рим" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

reversed\_text = ' '.join(word[::-1] for word in text.split())

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(reversed\_text)

### **Задача 11. Условие.**

В исходном текстовом файле замените все вхождения подстроки F на подстроку Z. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие подстроки F |
| Входные данные | F | Подстрока для замены | строка | любая |
| Входные данные | Z | Новая подстрока | строка | любая |
| Выходные данные | modified\_text | Текст с замененными подстроками | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "foo bar foo" |
| Входные данные | F | "foo" |
| Входные данные | Z | "baz" |
| Выходные данные | modified\_text | "baz bar baz" |

Листинг программы:  
F = input("Введите подстроку для замены: ")  
Z = input("Введите новую подстроку: ")

with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

modified\_text = text.replace(F, Z)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(modified\_text)

### **Задача 12. Условие.**

Для заданного символа определите, сколько раз он встречается во введенном тексте файла. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие символа |
| Входные данные | char | Символ для поиска | строка | одиночный символ |
| Выходные данные | count | Количество вхождений | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "hello world" |
| Входные данные | char | "o" |
| Выходные данные | count | 2 |

Листинг программы:  
char = input("Введите символ для поиска: ")

with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

count = text.count(char)  
print(f"Количество вхождений символа '{char}': {count}")

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(f"Символ '{char}' встречается {count} раз.")

### **Задача 13. Условие.**

Из текста, расположенного в файле, исключите группы символов, расположенных между круглыми скобками. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие скобок |
| Выходные данные | result | Текст без групп символов в скобках | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "Пример (текста) с (скобками)" |
| Выходные данные | result | "Пример с " |

Листинг программы:  
import re  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

result = re.sub(r'.∗?.\*?.∗?', '', text)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(result)

### **Задача 14. Условие.**

Из текста, расположенного в файле, исключите однобуквенные слова. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие слов |
| Выходные данные | result | Текст без однобуквенных слов | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "Я и ты" |
| Выходные данные | result | "ты" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

result = ' '.join(word for word in text.split() if len(word) > 1)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(result)

### **Задача 15. Условие.**

Из текста, расположенного в файле, удалите лишние пробелы, разделяющие слова.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | text | Текст из файла | строка | наличие пробелов |
| Выходные данные | result | Текст с одним пробелом между словами | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | text | "Это пример текста" |
| Выходные данные | result | "Это пример текста" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

result = ' '.join(text.split())

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(result)

### **Задача 16. Условие.**

Дан текстовый файл, содержащий целые числа. Найдите сумму четных элементов в каждой строке и допишите их в конец файла.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Числа из файла | список | целые числа |
| Выходные данные | even\_sum | Сумма четных элементов | целое |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | "1 2 3 4 5" |
| Выходные данные | even\_sum | 6 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

with open('output.txt', 'w', encoding='utf-8') as file:  
for line in lines:  
numbers = map(int, line.split())  
even\_sum = sum(num for num in numbers if num % 2 == 0)  
file.write(f"{line.strip()} Сумма четных: {even\_sum}\n")

### **Задача 17. Условие.**

Дан текстовый файл, содержащий целые числа. Найдите количество отрицательных элементов в каждой строке. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Числа из файла | список | целые числа |
| Выходные данные | negative\_count | Количество отрицательных чисел | целое |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | "-1 2 -3 4 5" |
| Выходные данные | negative\_count | 2 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

with open('output.txt', 'w', encoding='utf-8') as file:  
for line in lines:  
numbers = map(int, line.split())  
negative\_count = sum(1 for num in numbers if num < 0)  
file.write(f"{line.strip()} Количество отрицательных: {negative\_count}\n")

### **Задача 18. Условие.**

Дан текстовый файл, содержащий целые числа. Найдите номер минимального элемента в каждой строке. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Числа из файла | список | целые числа |
| Выходные данные | min\_index | Индекс минимального элемента | целое |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | "5 3 1 2" |
| Выходные данные | min\_index | 2 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

with open('output.txt', 'w', encoding='utf-8') as file:  
for line in lines:  
numbers = list(map(int, line.split()))  
min\_index = numbers.index(min(numbers))  
file.write(f"{line.strip()} Индекс минимального: {min\_index}\n")

### **Задача 19. Условие.**

Дан текстовый файл, содержащий целые числа. Найдите номер первого (слева направо) четного элемента в каждой строке. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Числа из файла | список | целые числа |
| Выходные данные | first\_even\_index | Индекс первого четного элемента | целое |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | "1 3 4 5" |
| Выходные данные | first\_even\_index | 2 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

with open('output.txt', 'w', encoding='utf-8') as file:  
for line in lines:  
numbers = list(map(int, line.split()))  
first\_even\_index = next((i for i, num in enumerate(numbers) if num % 2 == 0), -1)  
file.write(f"{line.strip()} Индекс первого четного: {first\_even\_index}\n")

### **Задача 20. Условие.**

Дан текстовый файл, содержащий целые числа. Найдите среднее арифметическое положительных чисел в файле. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Числа из файла | список | целые числа |
| Выходные данные | positive\_avg | Среднее арифметическое положительных чисел | вещественное |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | "-1 2 3 -4" |
| Выходные данные | positive\_avg | 2.5 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

positive\_numbers = [num for line in lines for num in map(int, line.split()) if num > 0]  
positive\_avg = sum(positive\_numbers) / len(positive\_numbers) if positive\_numbers else 0

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(f"Среднее арифметическое положительных чисел: {positive\_avg:.2f}")

### **Задача 21. Условие.**

Дан текстовый файл, содержащий целые числа. Найдите разность максимального и минимального чисел в файле. Результат запишите в другой файл.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Числа из файла | список | целые числа |
| Выходные данные | difference | Разность максимального и минимального чисел | целое |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | "1 5 3 9" |
| Выходные данные | difference | 8 |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

numbers = [num for line in lines for num in map(int, line.split())]  
difference = max(numbers) - min(numbers)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(f"Разность между максимальным и минимальным: {difference}")

### **Задача 22. Условие.**

Даны два текстовых файла. Запишите в третий только те строки, которые есть и в первом, и во втором файлах.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | lines1 | Строки первого файла | список строк |  |
| Входные данные | lines2 | Строки второго файла | список строк |  |
| Выходные данные | common\_lines | Общие строки | список строк |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | lines1 | "Привет\nКак дела?\n" |
| Входные данные | lines2 | "Как дела?\nХорошо\n" |
| Выходные данные | common\_lines | "Как дела?\n" |

Листинг программы:  
with open('file1.txt', 'r', encoding='utf-8') as file1, open('file2.txt', 'r', encoding='utf-8') as file2:  
lines1 = set(file1.readlines())  
lines2 = set(file2.readlines())

common\_lines = lines1.intersection(lines2)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.writelines(common\_lines)

### **Задача 23. Условие.**

Дан текстовый файл. Допишите в него следующие данные: количество строк, количество символов в каждой строке, количество цифр в каждой строке.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | lines | Строки файла | список строк |  |
| Выходные данные | line\_info | Информация о каждой строке | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | lines | "Привет 123\nКак дела?\n" |
| Выходные данные | line\_info | "Строка 1: символов 11, цифр 3\nСтрока 2: символов 10, цифр 0\n" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
lines = file.readlines()

with open('output.txt', 'a', encoding='utf-8') as file:  
for i, line in enumerate(lines, 1):  
char\_count = len(line.strip())  
digit\_count = sum(c.isdigit() for c in line)  
file.write(f"Строка {i}: символов {char\_count}, цифр {digit\_count}\n")

### **Задача 24. Условие.**

Дано некоторое конечное множество слов и текстовый файл. Разработайте программу исключения из текстового файла заданных слов.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | words | Список заданных слов | список строк |  |
| Входные данные | text | Текст из файла | строка |  |
| Выходные данные | result | Текст без заданных слов | строка |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | words | ["пример", "текст"] |
| Входные данные | text | "Это пример текста для примера" |
| Выходные данные | result | "Это для" |

Листинг программы:  
with open('input.txt', 'r', encoding='utf-8') as file:  
text = file.read()

words = ['пример', 'текст']  
result = ' '.join(word for word in text.split() if word not in words)

with open('output.txt', 'w', encoding='utf-8') as file:  
file.write(result)