МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Лабораторная работа № 1

по дисциплине «Программирование»

на тему «Символьные данные»

Выполнили

студенты группы М3О-125БВ-24

Вариант №4

Федоров А.И.,

Егоров А.В.

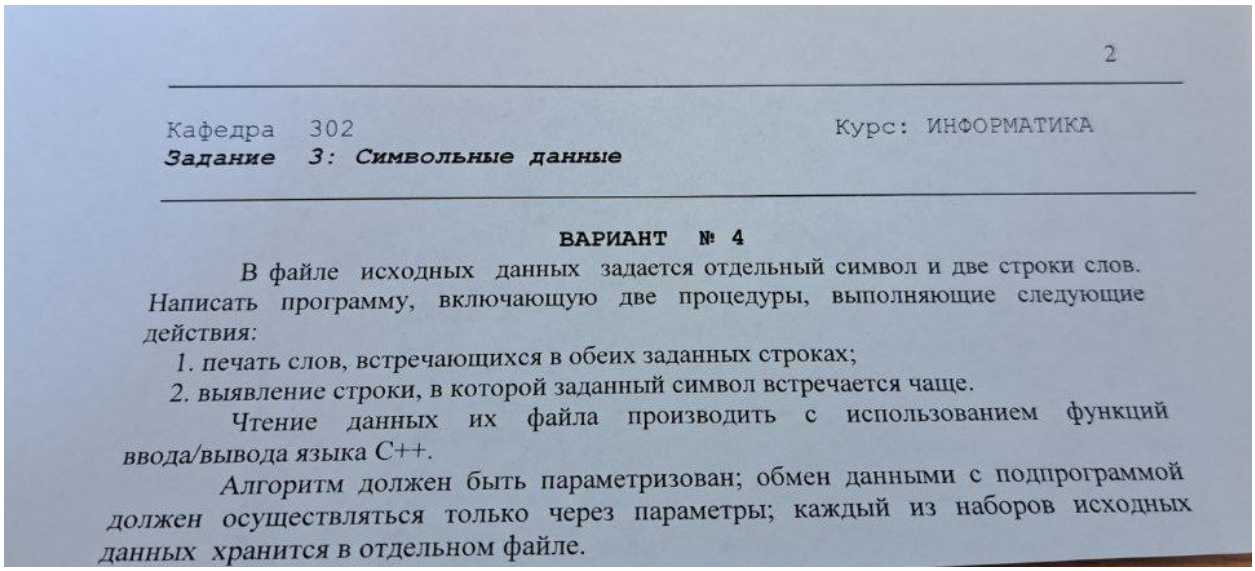
Приняли

ст. преп. каф. 304 Татаринкова Е.М.,

Москва
2025

Содержание:

Постановка задачи:



2

Кафедра 302 Курс: ИНФОРМАТИКА

Задание 3: Символьные данные

ВАРИАНТ № 4

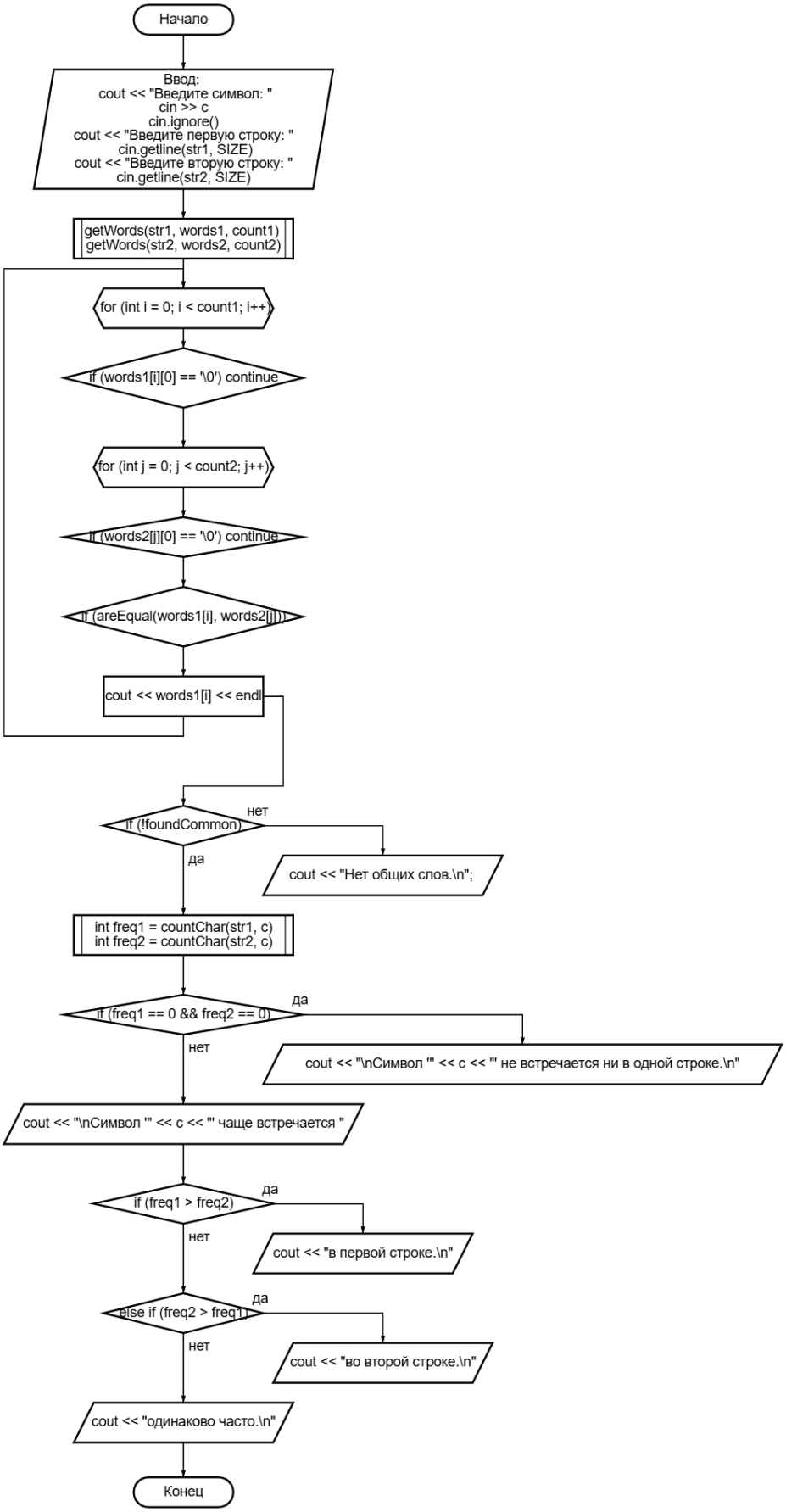
В файле исходных данных задается отдельный символ и две строки слов. Написать программу, включающую две процедуры, выполняющие следующие действия:

1. печать слов, встречающихся в обеих заданных строках;
2. выявление строки, в которой заданный символ встречается чаще.

Чтение данных их файла производить с использованием функций ввода/вывода языка C++.

Алгоритм должен быть параметризован; обмен данными с подпрограммой должен осуществляться только через параметры; каждый из наборов исходных данных хранится в отдельном файле.

Блок-схема:


Начало
Ввод:
cout << "Введите символ: "
cin >> c
cin.ignore()
cout << "Введите первую строку: "
cin.getline(str1, SIZE)
cout << "Введите вторую строку: "
cin.getline(str2, SIZE)
getWords(str1, words1, count1)
getWords(str2, words2, count2)
for (int i = 0; i < count1; i++)
if (words1[i][0] == '\0') continue
for (int j = 0; j < count2; j++)
if (words2[j][0] == '\0') continue
if (areEqual(words1[i], words2[j]))
cout << words1[i] << endl
if (!foundCommon)
нет
да
cout << "Нет общих слов.\n";
int freq1 = countChar(str1, c)
int freq2 = countChar(str2, c)
if (freq1 == 0 && freq2 == 0)
да
нет
cout << "\nСимвол '" << c << "' не встречается ни в одной строке.\n"
cout << "\nСимвол '" << c << "' чаще встречается "
if (freq1 > freq2)
да
нет
cout << "в первой строке.\n"
else if (freq2 > freq1)
да
нет
cout << "во второй строке.\n"
cout << "одинаково часто.\n"
Конец

Описание функции

Функция getWords

1. Назначение: разбивает строку на слова по пробелам

2. Прототип функции:

    void getWords(char str[], char words[][SIZE], int &count)

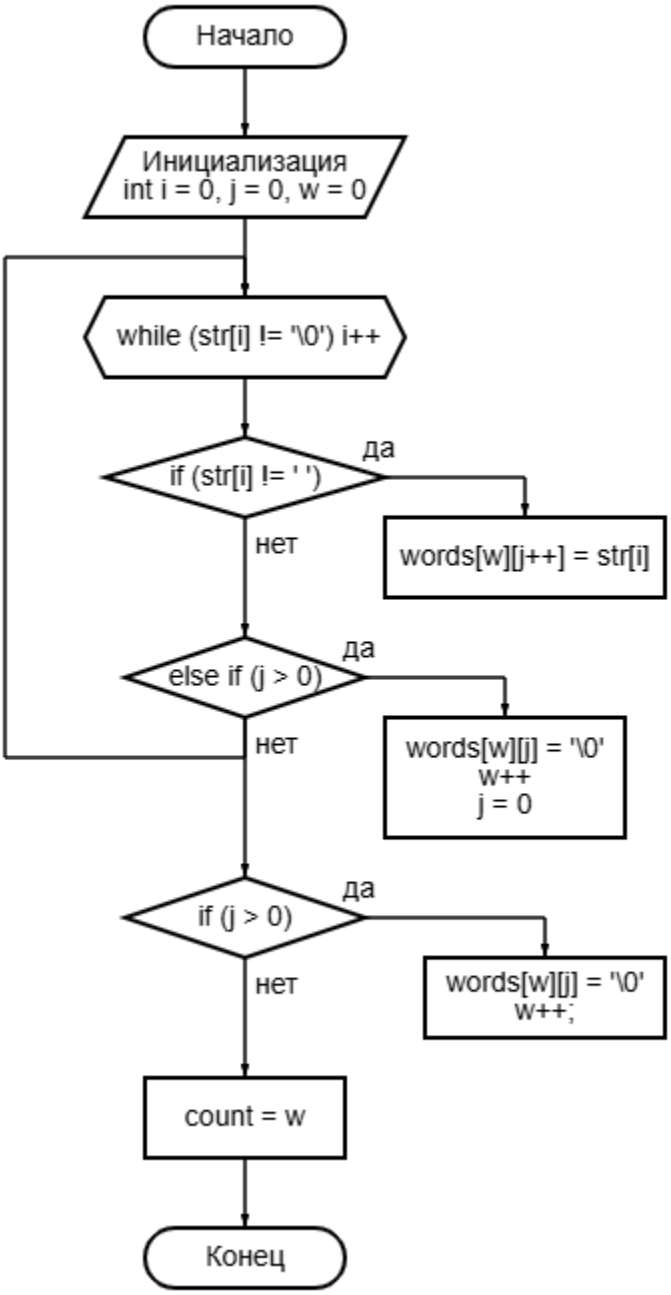
3. Обращение:

    getWords (str, words, count)

4. Описание параметров

Идентификатор	Тип	Назначение	Входной/ Выходной
getWords	void	Разбиенье строки на слова по пробелам	выходной
str	char	Входная строка	входной
words	char	Массив для хранения отдельных слов	входной
count	int&	Количество слов	выходной

5. Блок – схема функции


Начало
Инициализация
int i = 0, j = 0, w = 0
while (str[i] != '\0') i++
if (str[i] != ' ')
да
нет
words[w][j++] = str[i]
else if (j > 0)
да
нет
words[w][j] = '\0'
w++
j = 0
if (j > 0)
да
нет
words[w][j] = '\0'
w++;
count = w
Конец

Описание функции

Функция areEqual

1. Назначение: сравнивает два слова на равенство

2. Прототип функции:

    bool areEqual(char w1[], char w2[])

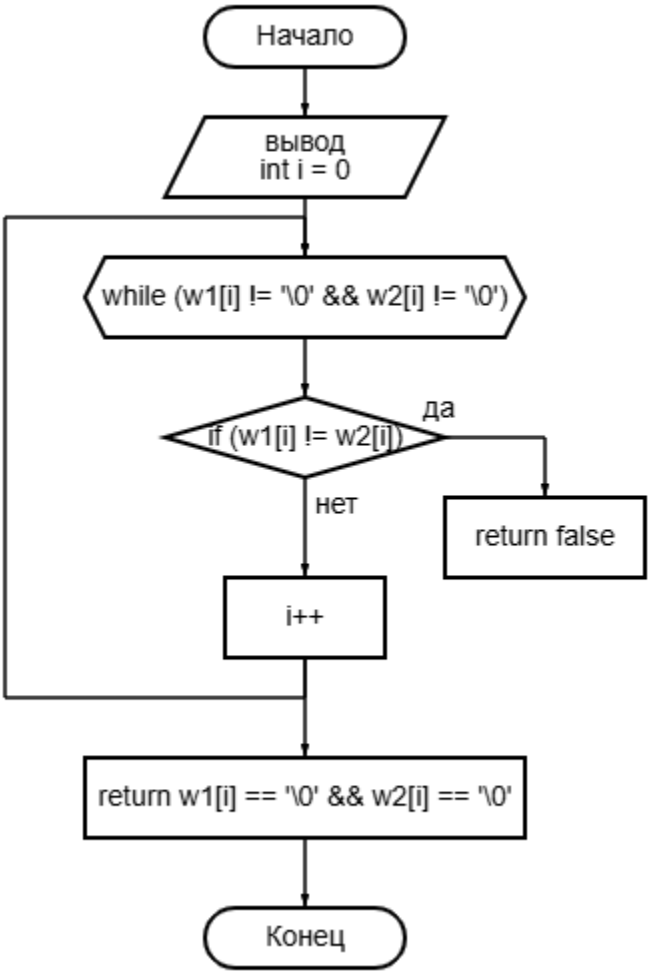
3. Обращение:

    areEqual (word1, word2)

4. Описание параметров:

Идентификатор	Тип	Назначение	Входной/ Выходной
areEqual	bool	Проверка слов на идентичность	выходной
w1	char	Первое слово	входной
w2	char	Второе слово	входной

5. Блок – схема функции


Начало
вывод
int i = 0
while (w1[i] != '\0' && w2[i] != '\0')
if (w1[i] != w2[i])
да
нет
return false
i++
return w1[i] == '\0' && w2[i] == '\0'
Конец

Описание функции

Функция countChar

1. Назначение: Считает количество вхождений символа в строке

2. Прототип функции:

```
int countChar(char str[], char c)
```

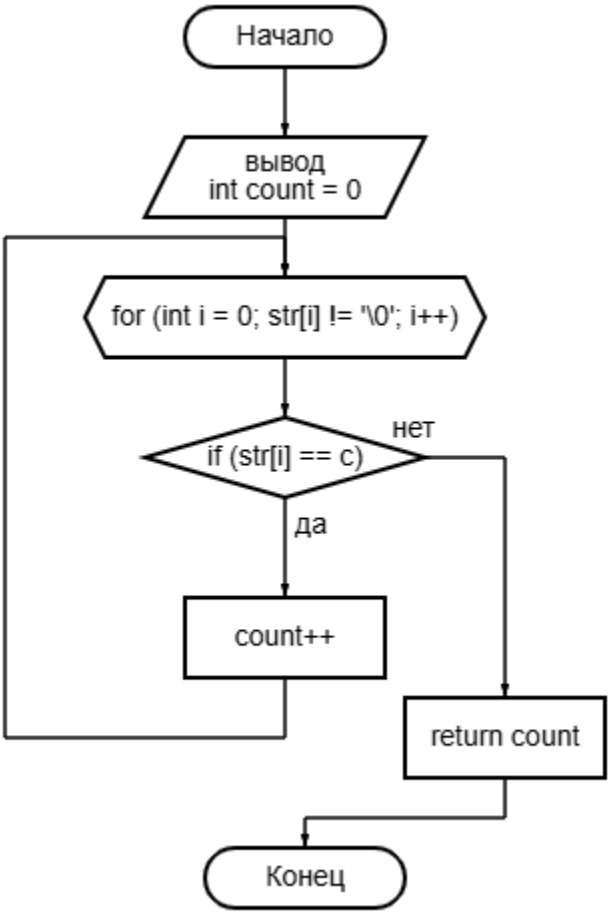
3. Обращение:

```
countChar (str, c)
```

4. Описание параметров

Идентификатор	Тип	Назначение	Входной/ Выходной
countChar	int	Подсчёт вхождений символа в строке	выходной
str	char	Строка для подсчёта вхождений символа	входной
c	char	Символ для проверки	входной

5. Блок – схема функции


Начало
вывод
int count = 0
for (int i = 0; str[i] != '\0'; i++)
if (str[i] == c)
нет
да
count++
return count
Конец

Код программы:

```
/*************************************************************
*                     КАФЕДРА № 304 1 КУРС                   *
*------------------------------------------------------------*
* Project Type  : GNU/Linux Console Application               *
* Project Name  : laba_1                                      *
* File Name     : main.cpp                                    *
* Language      : C/C++                                       *
* Programmer(s) : Егоров А.В (swrneko), Федоров А.И.          *
* Modifyed By   : Егоров А.В (swrneko)                        *
* Edited by     : Neovim, Visual Studio                       *
* OS            : Arch Linux, Windows 11                      *
* Github url    : https://github.com/swrneko/mai_shit.git     *
* Created       : 04/04/25                                    *
* Last Revision : 23/04/25                                    *
* Comment(s)    : Символьные данные                           *
*************************************************************/

#include <iostream>
using namespace std;

/************************
* Инициализация констант *
************************/
const int SIZE = 100;

/*******************
* Прототипы функций *
*******************/

// Разбивает строку на слова
// Параметры:
// - str: входная строка, содержащая слова, разделённые пробелами
// - words: двумерный массив для хранения отдельных слов
// - count: ссылка на переменную, в которую будет записано количество найденных слов
void getWords(char str[], char words[][SIZE], int &count);

// Сравнивает два слова на равенство
// Параметры:
// - w1: первое слово
// - w2: второе слово
// Возвращает true, если слова одинаковые, иначе false
bool areEqual(char w1[], char w2[]);

// Считает количество вхождений символа в строке
// Параметры:
// - str: строка, в которой выполняется подсчёт
// - c: символ, который нужно посчитать
// Возвращает количество вхождений символа c в строке str
int countChar(char str[], char c);

/******************
* Главная функция *
******************/

// --- Главная функция программы ---
int main() {
    char c;                      // Заданный символ
    char str1[SIZE], str2[SIZE]; // Две строки, введённые пользователем

    // Двумерные массивы для хранения слов из каждой строки
    char words1[20][SIZE], words2[20][SIZE];
```

```
    int count1 = 0, count2 = 0; // Количество слов в каждой строке

    // Ввод данных
    cout << "Введите символ: ";
    cin >> c;
    cin.ignore(); // Убираем символ новой строки после ввода символа

    cout << "Введите первую строку: ";
    cin.getline(str1, SIZE);

    cout << "Введите вторую строку: ";
    cin.getline(str2, SIZE);

    // Разделение строк на слова
    getWords(str1, words1, count1);
    getWords(str2, words2, count2);

    // --- Поиск и вывод общих слов ---
    bool foundCommon = false; // флаг наличия общих слов

    cout << "\nОбщие слова:\n";
    for (int i = 0; i < count1; i++) {
        if (words1[i][0] == '\0') continue; // пропустить пустые

        for (int j = 0; j < count2; j++) {
            if (words2[j][0] == '\0') continue;

            if (areEqual(words1[i], words2[j])) {
                cout << words1[i] << endl;
                foundCommon = true;
                break;
            }
        }
    }

    if (!foundCommon) {
        cout << "Нет общих слов.\n";
    }

    // --- Подсчёт количества символов и вывод результата ---
    int freq1 = countChar(str1, c);
    int freq2 = countChar(str2, c);

    if (freq1 == 0 && freq2 == 0) {
        cout << "\nСимвол '" << c << "' не встречается ни в одной строке.\n";
    } else {
        cout << "\nСимвол '" << c << "' чаще встречается ";
        if (freq1 > freq2)
            cout << "в первой строке.\n";
        else if (freq2 > freq1)
            cout << "во второй строке.\n";
        else
            cout << "одинаково часто.\n";
    }

    return 0;
}

/**********
* Функции *
**********/

// Разбивает строку на слова по пробелам
void getWords(char str[], char words[][SIZE], int &count) {
```

```
    int i = 0, j = 0, w = 0;
    count = 0;

    while (str[i] != '\0') {
        if (str[i] != ' ') {
            words[w][j++] = str[i]; // добавляем символ в слово
        } else if (j > 0) {
            // Если j > 0, значит было накоплено хотя бы одно слово
            words[w][j] = '\0'; // закрываем слово
            w++;
            j = 0;
        }
        i++;
    }

    // Последнее слово (если было)
    if (j > 0) {
        words[w][j] = '\0';
        w++;
    }

    count = w; // сохраняем количество слов
}

// Сравнивает два слова символ за символом
bool areEqual(char w1[], char w2[]) {
    int i = 0;
    while (w1[i] != '\0' && w2[i] != '\0') {
        if (w1[i] != w2[i])
            return false; // Нашли несовпадение
        i++;
    }

    // Возвращаем true только если оба слова закончились одновременно
    return w1[i] == '\0' && w2[i] == '\0';
}

// Подсчёт количества вхождений символа в строку
int countChar(char str[], char c) {
    int count = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == c)
            count++;
    }
    return count;
}
```

Тесты:

1. Нормальный тест:

```
Введите символ: a
Введите первую строку: some words without any mean
Введите вторую строку: another words but now maybe with mean

Общие слова:
words
mean

Символ 'a' чаще встречается во второй строке.
```

2. Тест на одинаковые слова в обеих строчках:

```
Введите символ: s
Введите первую строку: some words
Введите вторую строку: some words

Общие слова:
some
words

Символ 's' чаще встречается одинаково часто.
```

3. Тест с без одинаковых слов в обеих строчках:

```
Введите символ: g
Введите первую строку: g test name
Введите вторую строку: t mom bob

Общие слова:
Нет общих слов.

Символ 'g' чаще встречается в первой строке.
```

4. Тест на одинаковое количество повторяющихся символов в обеих строчках:

```
Введите символ: s
Введите первую строку: name home test
Введите вторую строку: lame none nest

Общие слова:
Нет общих слов.

Символ 's' чаще встречается одинаково часто.
```

5. Тест на отсутствие повторяющихся символов в обеих строчках:

```
Введите символ: j
Введите первую строку: lame fame same nose
Введите вторую строку: bob summer lost

Общие слова:
Нет общих слов.

Символ 'j' не встречается ни в одной строке.
```

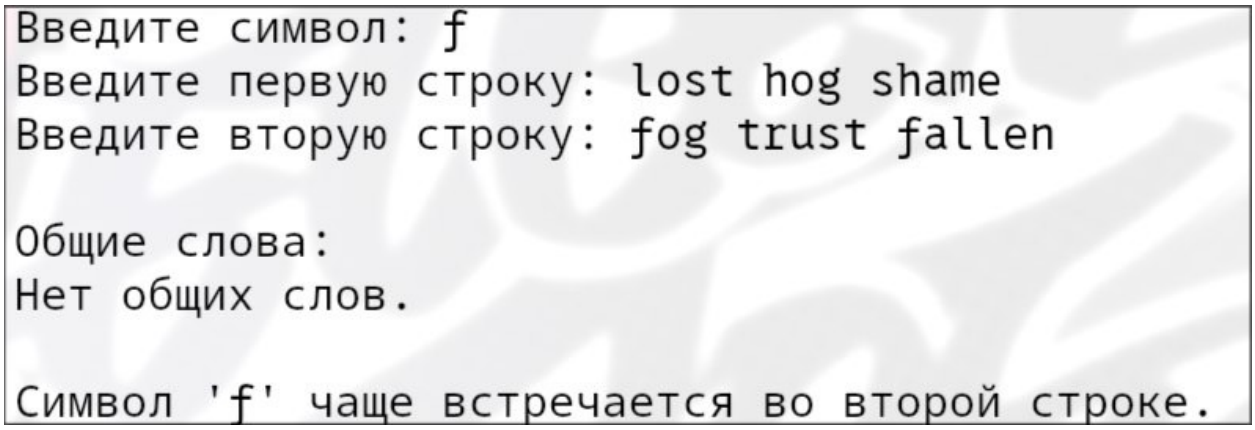
6. Тест на повторяющийся символ в первой строчке:

```
Введите символ: j
Введите первую строку: jackob bob name
Введите вторую строку: losten find test

Общие слова:
Нет общих слов.

Символ 'j' чаще встречается в первой строке.
```

7. Тест на повторяющийся символ во второй строчке:



Вывод по работе:

Разработка программы завершена на том основании, что:

1. Полученный результаты совпали с ожидаемыми;
2. Считаем набор тестов полным.