МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №6

по дисциплине: Основы программирования тема: «Использование функций при работе со строками»

Выполнил: ст. группы ПВ-201 Машуров Дмитрий Русланович

Проверил:

Притчин Иван Сергеевич Брусенцева Валентина

Станиславовна

Белгород 2021 г.

#### ЛАБОРАТОРНАЯ РАБОТА № 6

#### Использование функций при работе со строками

#### Цель работы: получение навыков работы со строками. Закрепление навыков работы с функциями.

#### Задания для подготовки к работе

#### Изучить организацию работы со строками.

#### Изучить стандартные функции для работы со строками в языке Си.

#### Разработать алгоритм и составить программу для решения задачи соответствующего варианта таким образом, чтобы решение каждой подзадачи описывалось функцией. Если не указано иначе, словом считать последовательность символов, не содержащую «пустых» символов (символов с кодами меньшими 33). Буквами считать буквы латинского алфавита.

#### Если в стандартной библиотеке есть функция для решения выделенной подзадачи, то описать свою функцию с таким же заголовком и назначением и не использовать стандартную.

#### Подобрать наборы тестовых данных.

**Задание варианта №17**

#### В данной строке соседние слова разделены запятыми. Определить количество слов-палиндромов, которые начинаются с буквы “а”.

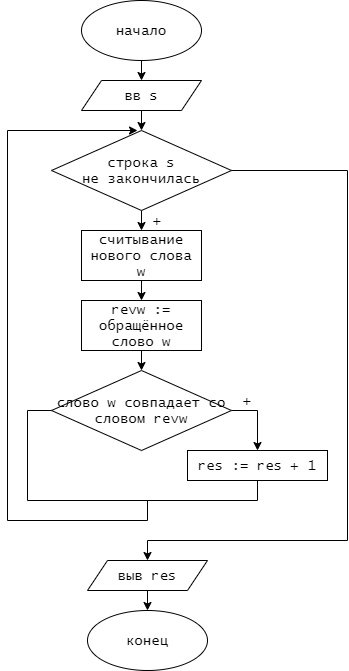
**Выполнение работы:**

1. **Описание алгоритма и выделение подзадач**

Исходя из условия задачи, будем брать слово из строки, обращать его и сравнивать с исходным. Если слова будут одинаковы и исходное будет начинать с “а”, то будем включать его в подсчёт палиндромов, начинающихся с “a”.

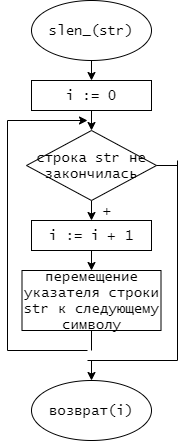
Выделим следующие подзадачи:

1. Обращение строки
2. Определение равенства строк
3. Возвращение кол-ва палиндромов, начинающихся с «а», в исходной строке
4. **Блок-схема с укрупнёнными блоками**



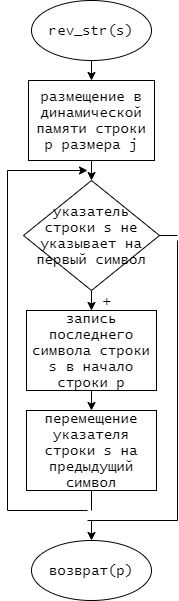
1. **Спецификации функций**
   1. Нахождение длины строки
      1. Заголовок: int slen(char \*str)
      2. Назначение: возвращает длину строки str

Блок-схема:



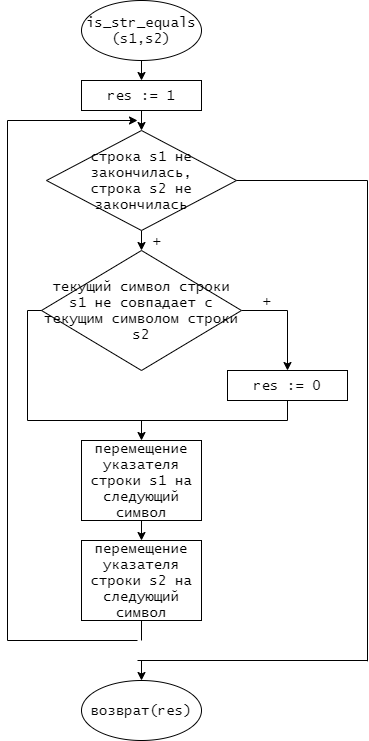
1. Обращение строки
   1. Выделение подзадач
      1. Нахождение длины строки
   2. Заголовок: char \* rev\_str(char \*s)
   3. Назначение: возвращает обращенную строку s

Блок-схема:



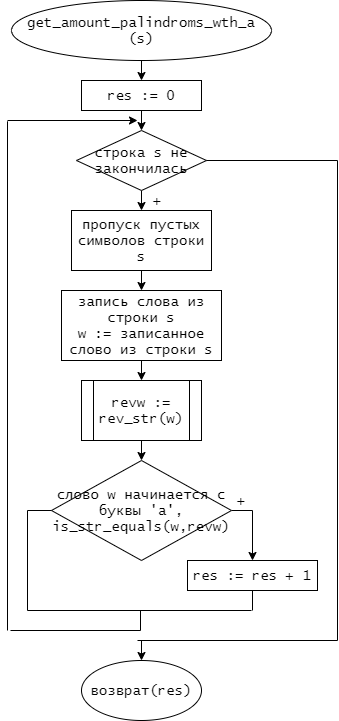
1. Определение равенства строк
   1. Заголовок: int is\_str\_equals(char \*s1, char \*s2)
   2. Назначение: возвращает «истину» если строки s1 и s2 одинаковые, иначе – «ложь»

Блок-схема:



1. Возвращение кол-ва палиндромов, начинающихся с «а», в исходной строке
   1. Заголовок: size\_t get\_amount\_palindroms\_wth\_a(char \*s)
   2. Назначение: возвращает кол-во палиндромов, начинающихся с буквы “a”, строки s

Блок-схема:



1. **Тестовые данные**

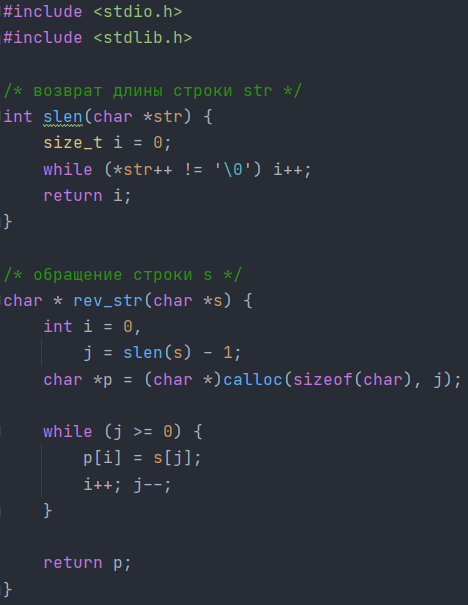
|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Вход** | **Выход** |
| **1** | “afdf, gerr, awwa, afa, allld” | 2 |
| **2** | “awaw, wawa, wwaaww, dlld, oeeeeeo” | 0 |

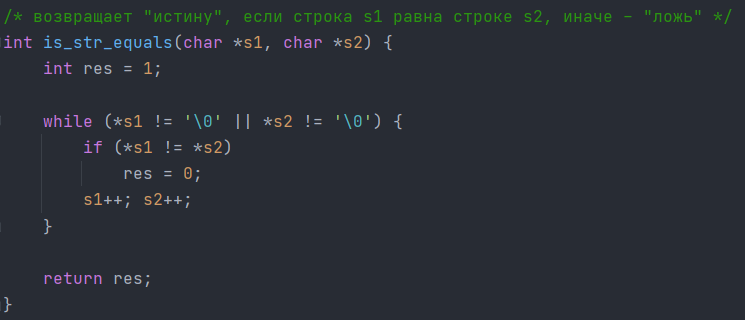
1. **Текст программы**

#### #include <stdio.h> #include <stdlib.h> /\* возврат длины строки str \*/ int slen(char \*str) { size\_t i = 0; while (\*str++ != '\0') i++; return i; } /\* обращение строки s \*/ char \* rev\_str(char \*s) { int i = 0, j = slen(s) - 1; char \*p = (char \*)calloc(sizeof(char), j); while (j >= 0) { p[i] = s[j]; i++; j--; } return p; } /\* возвращает "истину", если строка s1 равна строке s2, иначе - "ложь" \*/ int is\_str\_equals(char \*s1, char \*s2) { int res = 1; while (\*s1 != '\0' || \*s2 != '\0') { if (\*s1 != \*s2) res = 0; s1++; s2++; } return res; } /\* возвращает кол-во палиндромов, начинающихся с буквы 'a', строки s \*/ size\_t get\_amount\_palindroms\_wth\_a(char \*s) { size\_t res = 0; while (\*s != '\0') { while (\*s == ' ' || \*s == ',') s++; char w[255]; size\_t i = 0; while (\*s != ',' && \*s != '\0') { w[i] = \*s; s++; i++; } w[i] = '\0'; char \*revw = rev\_str(w); if (\*w == 'a' && is\_str\_equals(w, revw)) res++; } return res; } int main() { char s[255]; gets(s); size\_t res = get\_amount\_palindroms\_wth\_a(s); printf("Result: %d palindroms with \"a\"", res); }

1. **Результаты работы:**

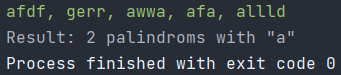
*Программа:*



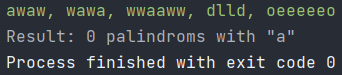




*Пример №1*



*Пример №2*



1. **Анализ ошибок**

* нет