

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

1. Изучить организацию работы со строками.
2. Изучить стандартные функции для работы со строками в языке Си.
3. 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.
4. Если в стандартной библиотеке есть функция для решения выделенной подзадачи, то описать свою функцию с таким же заголовком и назначением и не использовать стандартную.
5. Подобрать наборы тестовых данных.

Задание варианта №17

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

Выполнение работы:

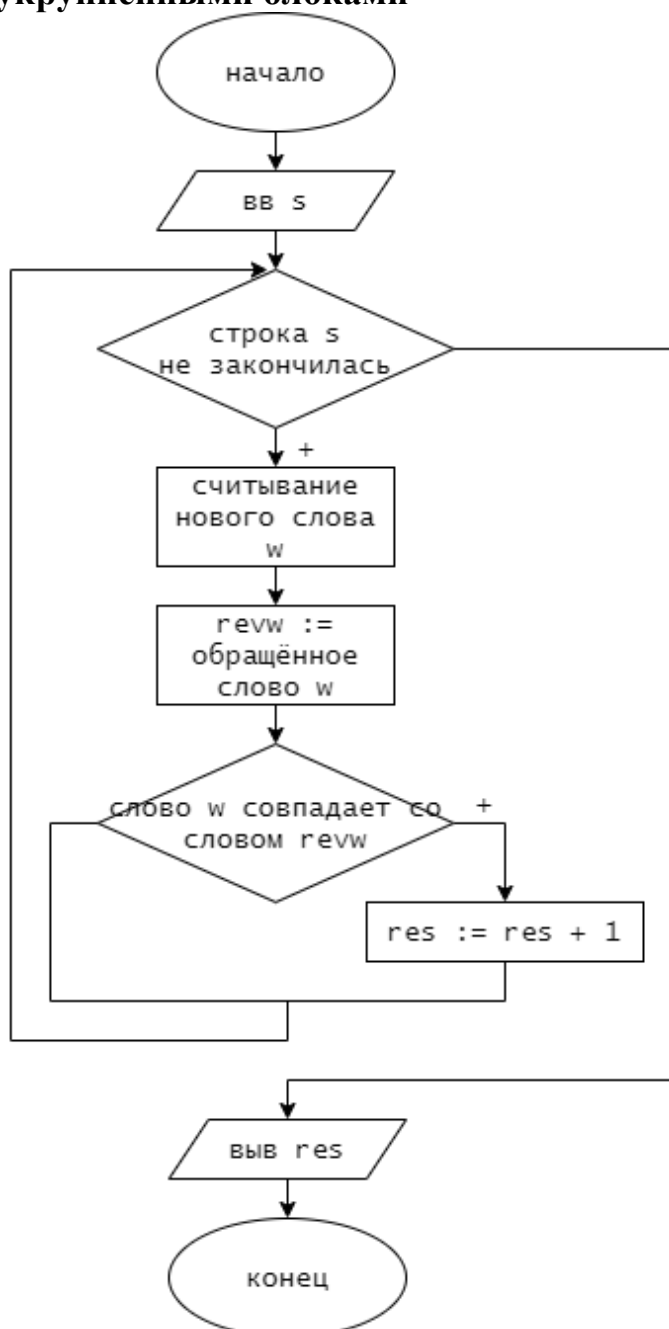
1. Описание алгоритма и выделение подзадач

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

Выделим следующие подзадачи:

- 1) Обращение строки
- 2) Определение равенства строк
- 3) Возвращение кол-ва палиндромов, начинающихся с «а», в исходной строке

2. Блок-схема с укрупнёнными блоками



3. Спецификации функций

1. Нахождение длины строки

а. Заголовок: `int slen(char *str)`

б. Назначение: возвращает длину строки `str`

Блок-схема:



2. Обращение строки

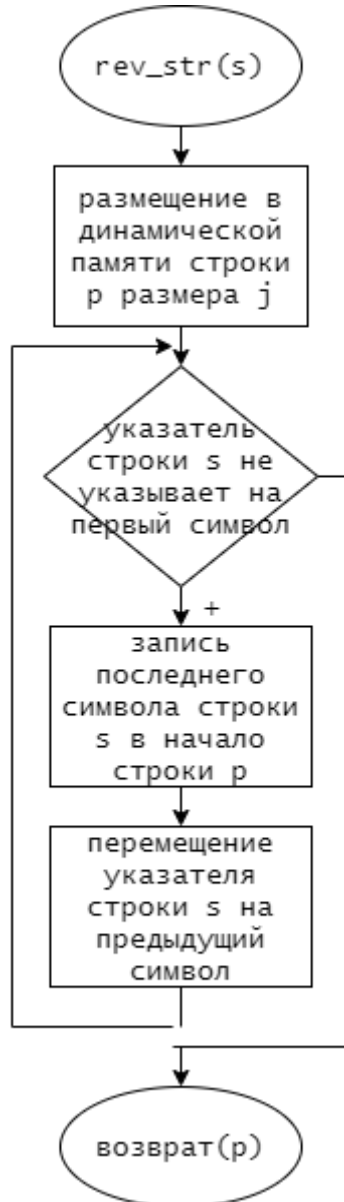
1. Выделение подзадач

а. Нахождение длины строки

2. Заголовок: `char * rev_str(char *s)`

3. Назначение: возвращает обращенную строку `s`

Блок-схема:

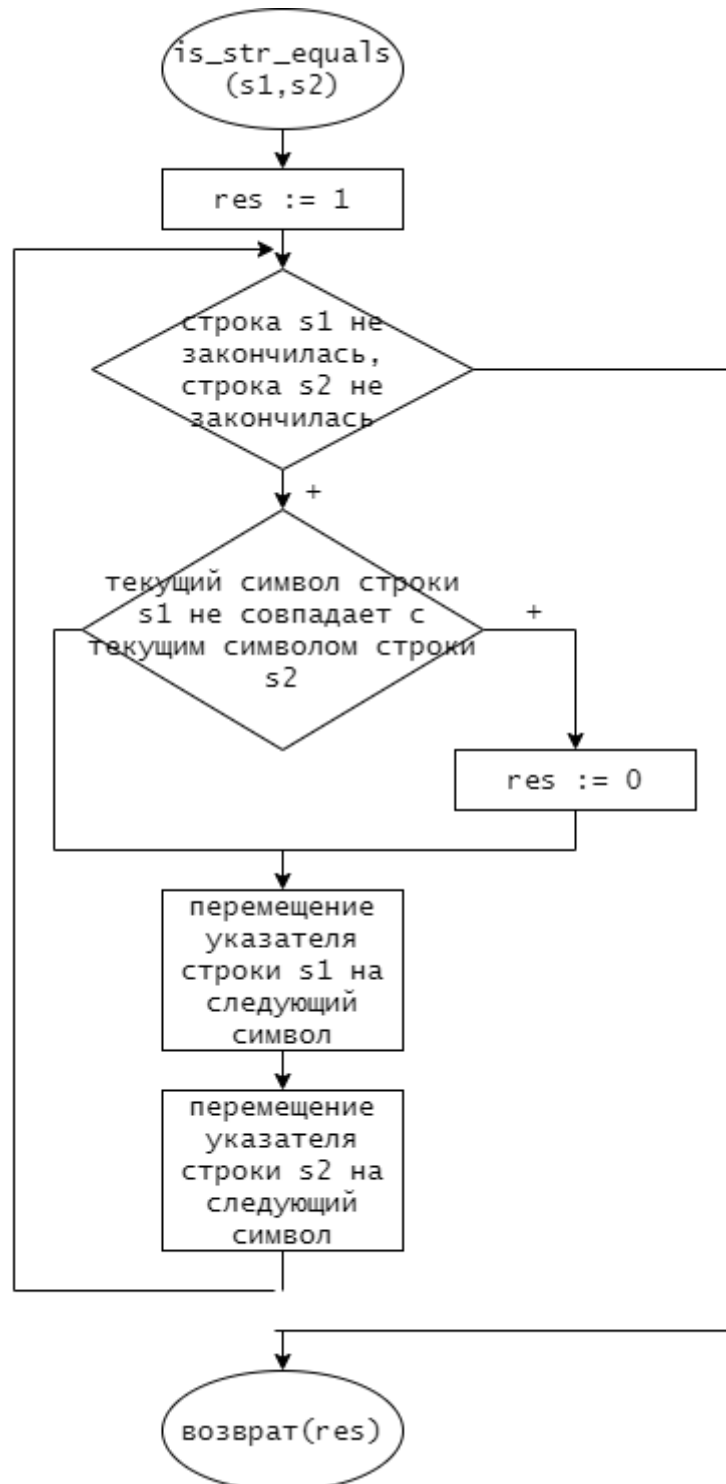


3. Определение равенства строк

1. Заголовок: `int is_str_equals(char *s1, char *s2)`

2. Назначение: возвращает «истину» если строки `s1` и `s2` одинаковые, иначе – «ложь»

Блок-схема:



4. 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 «а», строки `s`

Блок-схема:



5. Тестовые данные

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

6. 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);
}

```

7. Результаты работы:

Программа:

```
#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\\n\", res);
}

```

Пример №1

```

afdf, gerr, awwa, afa, allld
Result: 2 palindroms with "a"
Process finished with exit code 0

```

Пример №2

```

awaw, wawa, wwaaww, dllld, oeeeeeo
Result: 0 palindroms with "a"
Process finished with exit code 0

```

8. Анализ ошибок

- нет