**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

отчет

**по лабораторной работе № 5**

**по дисциплине «Программирование»**

Тема: «Обработка строк»

|  |  |  |
| --- | --- | --- |
| Студент гр. 3311 |  | Шарпинский Д. А. |
| Преподаватель |  | Хахаев И. А. |

Санкт-Петербург

2023

**Цель работы.**

Научиться работать со строками в языке С.

**Задание (вариант 2)**

Ввести строку текста, которая состоит из слов и произвольного количества символов-разделителей до и после слов. Массив символов-разделителей вводится после ввода строки. Количество символов-разделителей задается. Подсчитать и вывести количество слов максимальной длины в строке.

**Постановка задачи и описание решения**

Поставлена задача считать строку, количество символов разделителей и строку самих символов-разделителей, обработать введенные данные и вывести количество слов максимальной длины. Для этого напишем программу.

Чтение строк осуществляется с помощью функции fgets.

Строки ограничены по длине в 500 символов.

Строка символов-разделителей обрезается по введённому количеству символов-разделителей.

Далее происходит обработка с помощью двух функций: is\_delimiter и count\_words.

is\_delimiter принимает два аргумента: символ и массив символов разделителей, возвращает 0 или 1 – в зависимости от того, является ли символ разделителем (0 – не является, 1 – является). Проверка происходит путём сравнения символа с каждым элементом массива разделителей.

count\_words принимает два аргумента: исходную строку и массив символов-разделителей. Сначала проверяется, есть ли хотя бы один символ не разделитель в исходной строке, если нет – функция завершает работу и возвращает 0. В противном случае идет проверка каждого символа строки: если это не разделитель – увеличивается длина текущего слова, иначе происходит сравнение длины текущего слова с максимальной, обновление максимальной длины, если текущая длина оказалась больше, обновление счетчика слов, а также обнуление длины текущего слова. Так как обработка происходит в цикле до тех пор, пока не будет получен символ ‘\0’, то появляется необходимость проверки последнего слова вне цикла. После чего функция возвращает количество слов.

**Описание переменных**

**Функция main():**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | s | char[500] | Хранение введенной строки |
| 2 | d | char[500] | Хранение введенных символов-разделителей |
| 3 | n | int | Количество символов разделителей |
| 4 | count | int | Количество слов максимальной длины |

**Функция is\_delimiter():**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | ch | char | Проверяемый символ |
| 2 | delimiters | char[500] | Массив символов-разделителей |
| 3 | i | int | Индекс в цикле for |
| 4 | result | int | Результат работы функции |

**Функция count\_words():**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | s | char[500] | Проверяемая строка |
| 2 | delimiters | char[500] | Массив символов-разделителей |
| 3 | i | int | Индекс в цикле for |
| 4 | max\_length | int | Максимальная длина слова |
| 5 | current\_length | int | Длина текущего слова |
| 6 | check | int | Хранение результатов логических проверок |
| 7 | count | int | Счётчик слов максимальной длины |

**Контрольные примеры**

**Пример 1:**

s: a b cc cc

n: 1

delimiter characters:

delimiter characters after trimming to length n (between ||): | |

count: 2

**Пример 2:**

s: 12

n: 2

delimiter characters: 12

delimiter characters after trimming to length n (between ||): |12|

count: 0

**Пример 3:**

s: ,,,, ,, , , ,, ,,,

n: 10

delimiter characters: ,

delimiter characters after trimming to length n (between ||): |,

|

count: 3

**Пример 4:**

s: main\_name\_call\_whole

n: 1

delimiter characters: \_who\_call\_man

your delimiter characters after trimming to length n (between ||): |\_|

count: 1

**Пример 5:**

s:

n: 1

delimiter characters:

your delimiter characters after trimming to length n (between ||): |

|

count: 0

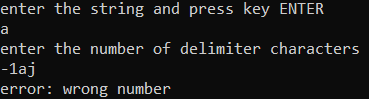
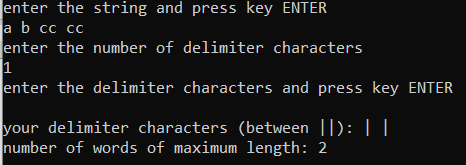
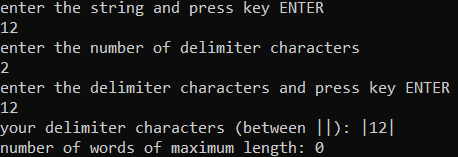
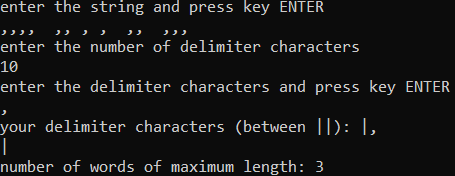
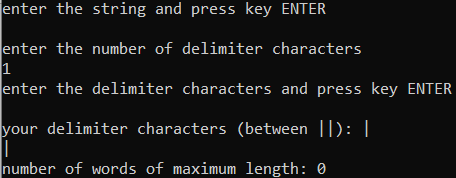
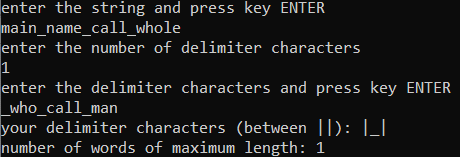
**Пример 6:**

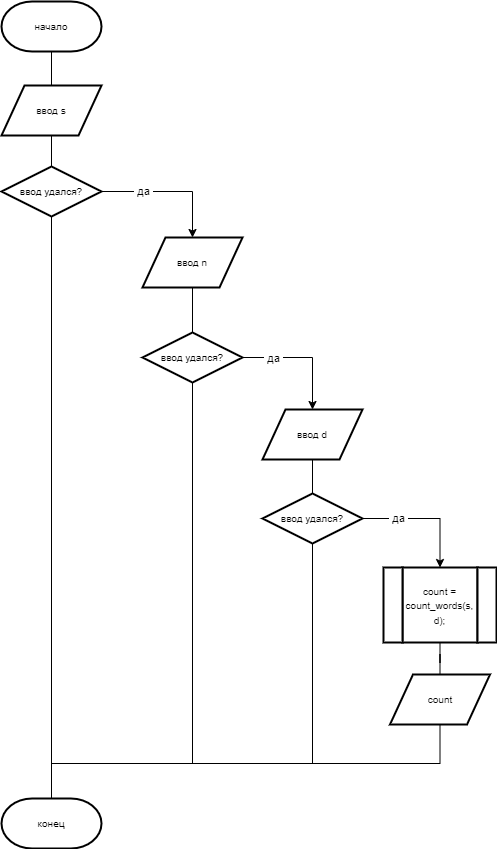
s: a

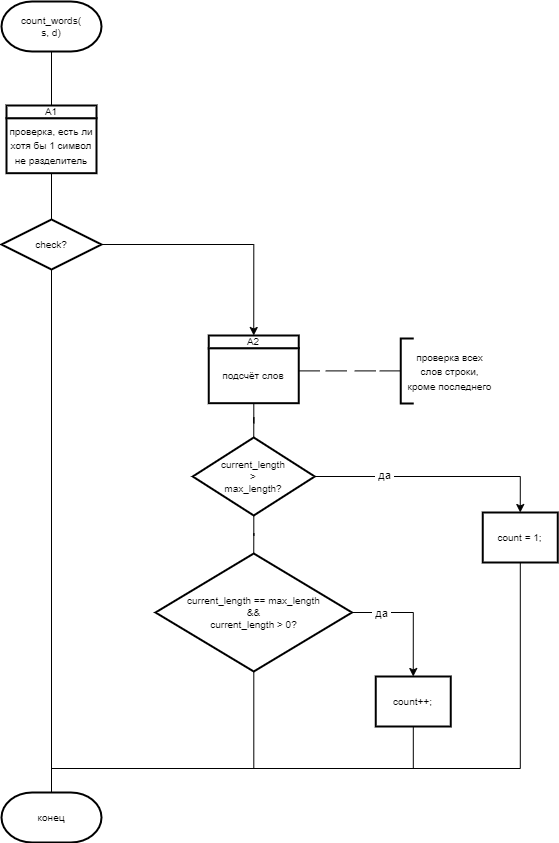
n: -1aj

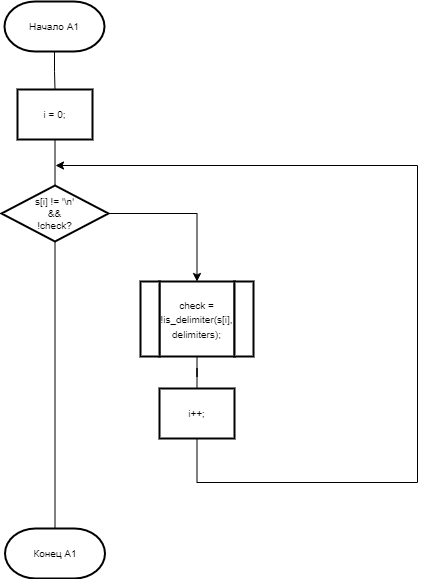
error: wrong number

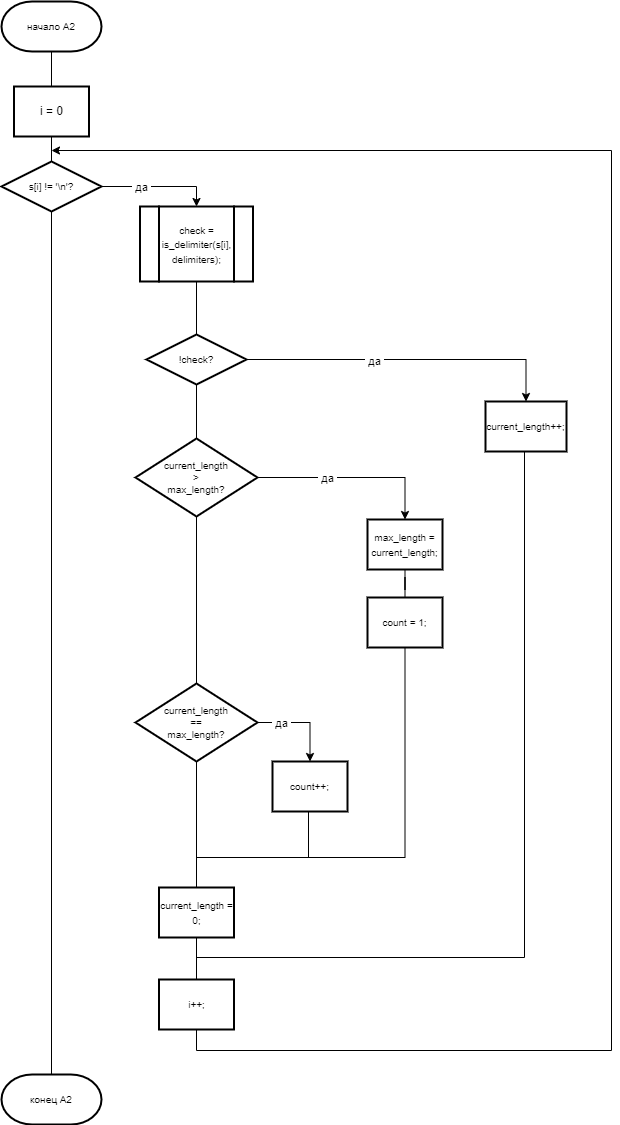
**Примеры выполнения программы**

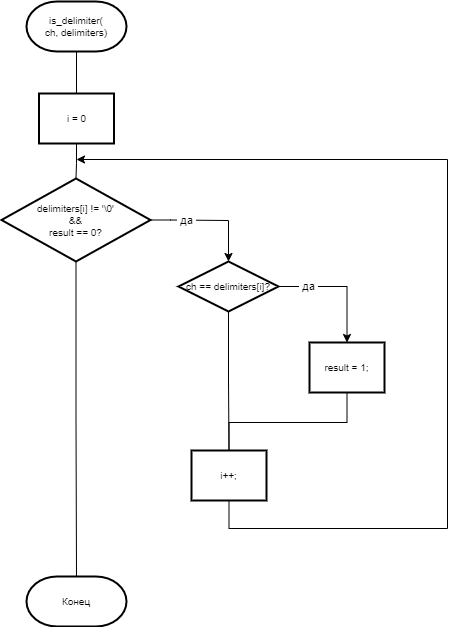
****

**Блок-схема**

****

****

****

****

**Текст программы**

#include <stdio.h>

#include <string.h>

#define MAX\_LENGTH 500

int is\_delimiter(char ch, const char \*delimiters);

int count\_words(const char \*s, const char \*delimiters);

int main() {

char s[MAX\_LENGTH], d[MAX\_LENGTH];

int n, count;

printf("enter the string and press key ENTER\n");

if (fgets(s, MAX\_LENGTH, stdin) != NULL) {

printf("enter the number of delimiter characters\n");

if (scanf("%d", &n) && n > 0) {

getchar();

printf("enter the delimiter characters and press key ENTER\n");

if (fgets(d, n + 1, stdin) != NULL) {

printf("your delimiter characters (between ||): |%s|\n", d);

count = count\_words(s, d);

printf("number of words of maximum length: %d\n", count);

} else {

printf("error: failed to input string\n");

}

} else {

printf("error: wrong number");

}

} else {

printf("error: failed to input string\n");

}

return 0;

}

int is\_delimiter(char ch, const char \*delimiters) {

int result = 0, i;

for (i = 0; delimiters[i] != '\0' && !result; i++) {

if (ch == delimiters[i]) {

result = 1;

}

}

return result;

}

int count\_words(const char \*s, const char \*delimiters) {

int max\_length = 0, current\_length = 0, count = 0, check = 0, i;

for (i = 0; s[i] != '\n' && !check; i++) {

check = !is\_delimiter(s[i], delimiters);

}

if (check) {

for (i = 0; s[i] != '\n'; i++) {

check = is\_delimiter(s[i], delimiters);

if (!check) {

current\_length++;

} else {

if (current\_length > max\_length) {

max\_length = current\_length;

count = 1;

} else if (current\_length == max\_length) {

count++;

}

current\_length = 0;

}

}

if (current\_length > max\_length) {

count = 1;

} else if (current\_length == max\_length && current\_length > 0) {

count++;

}

}

return count;

}

**Выводы.**

Были получены навыки работы со строками в языке С.