**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по курсовой работе**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 9301 |  | Служевская А.С. |
| Преподаватель |  | Рыжов Н.Г. |

Санкт-Петербург

2019

**Постановка задачи**

**лоуацддлраыдлваыдлоавылодфваылодвфаылодвфылод**

Составить функцию для разделения текста, заданного строкой литер, на отдельные слова и подсчета числа слов. Под словом понимается последовательность литер, отличающаяся от пробела, ограниченная слева началом строки или пробелом и справа — пробелом, знаком препинания или концом строки.

**Схема алгоритма решения**

На рисунке 1 представлена блок-схема решения задачи по решению задачи.

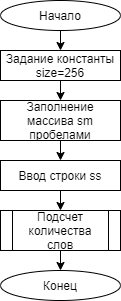
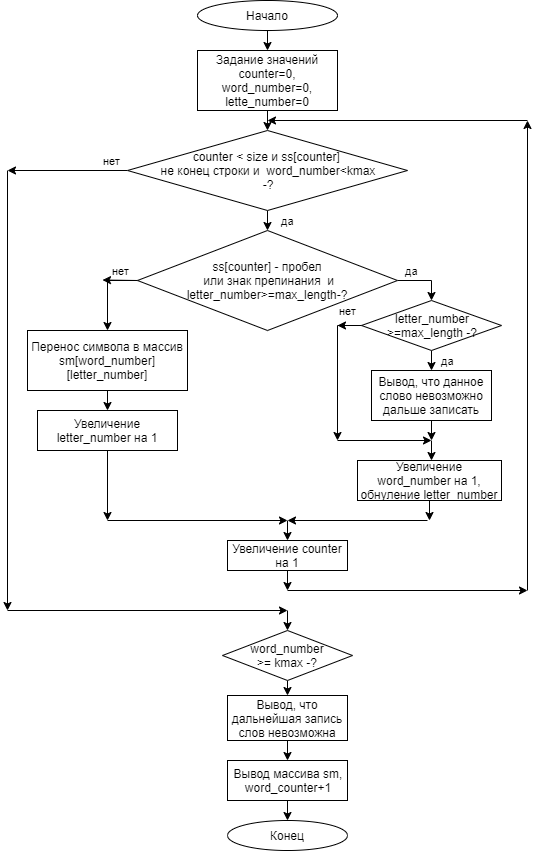
Рисунок 2. Функция kwords()

Рисунок . Решение задачи



**Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <iostream>

#include <conio.h>

#include <clocale>

#include <windows.h>

#include <math.h>

#include <stdlib.h>

#include <cmath>

#include <math.h>

using namespace std;

#define Russian 1251 //создание макросов

#define kmax 100

#define max\_length 30

int kwords(char \*ss, char sm[kmax][max\_length], int size){ //функция для разделения на слова

int counter = 0, word\_number = 0, letter\_number = 0; //задание счетчиков

while (counter < size && ss[counter] != '\0' && word\_number < kmax) { //перебор массива и проверка, что строка не закончилась и что не превышено количество слов

if (ss[counter] != ' ' && //проверка на знаки препинания

ss[counter] != ',' &&

ss[counter] != '.' &&

ss[counter] != '!' &&

ss[counter] != '?' &&

ss[counter] != ';' &&

ss[counter] != ':' &&

letter\_number < max\_length) { //проверка на длину слова

sm[word\_number][letter\_number] = ss[counter]; //перенос символа в массив со словами

letter\_number++; //увеличение индекса буквы в массиве со словами

}

else {

if (letter\_number >= max\_length)

cout << "Слово не может быть записано корректно. Длина слова больше возможной" << endl;

word\_number++; //увеличение индекса слова в массиве со словами

letter\_number = 0; //обнуление индекса буквы

}

counter++;

}

if (word\_number >= kmax)

cout << "Дальнейшая запись слов невозможна. Превышено число слов." << endl;

for (int i = 0; i < kmax; i++) {

for (int j = 0; j < max\_length; j++) {

if (sm[i][j] != ' ') //проверка, что символ не пустой

cout << sm[i][j]; //вывод слова из массива слов

}

if (sm[i][0] != ' ') { //проверка, что строка не пустая

cout << endl;

}

}

cout << "Всего в строке " << word\_number+1 << " слов"; //вывод счетчика слов

return 0;

}

int main()

{

SetConsoleCP(Russian);

SetConsoleOutputCP(Russian);

const int size = 256; //задание константы размера массива

char ss[size], sm[kmax][max\_length]; //задание массивов

for (int i = 0; i < kmax; i++) { // заполнение массива со словами пробелами

for (int j = 0; j < max\_length; j++) {

sm[i][j] = ' ';

}

}

cout << "Введите строку" << endl;

gets\_s(ss); //ввод строки

kwords(ss, sm, size); //вызов функции

}

**Описание программы**

Программа получает на вход строку. Далее программа заполняет массив под разделенные слова пробелами, после чего вызывает функцию для разделения строки на слова. В функцию передаются массив со строкой, массив под слова, размер массива со строкой. Далее программа проверяет каждый элемент массива строки. Если этот элемент — знак препинания или пробел, то программа проверяет дальше, иначе переносит элемент в массив со словами. После этого программа берет следующий элемент. После этого программа выводит полученные слова и их количество.

**Руководство пользователя**

От пользователя требуется ввести строку. Длина строки не должна превышать 256.

**Пути дальнейшего улучшения программы**

* Улучшить пользовательский интерфейс
* Добавить возможность задавать длину строки
* Добавить возможность изменять максимальное количество слов и максимальную длину пользователем