**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**Отчет По Дисциплине**

**«Алгоритмизация и программирование»**

**Лабораторная работа №1**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-115

Клепиков Степан Даниилович

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ФИО)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ФИО)

Москва 2022 г.

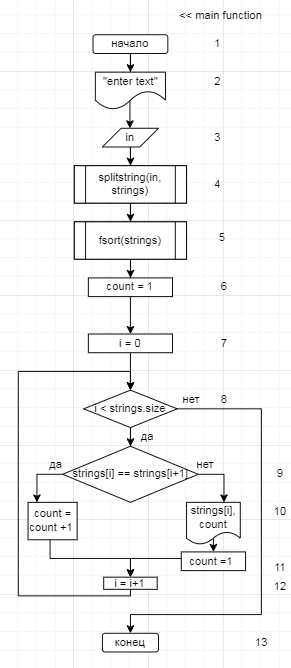
**Задание**

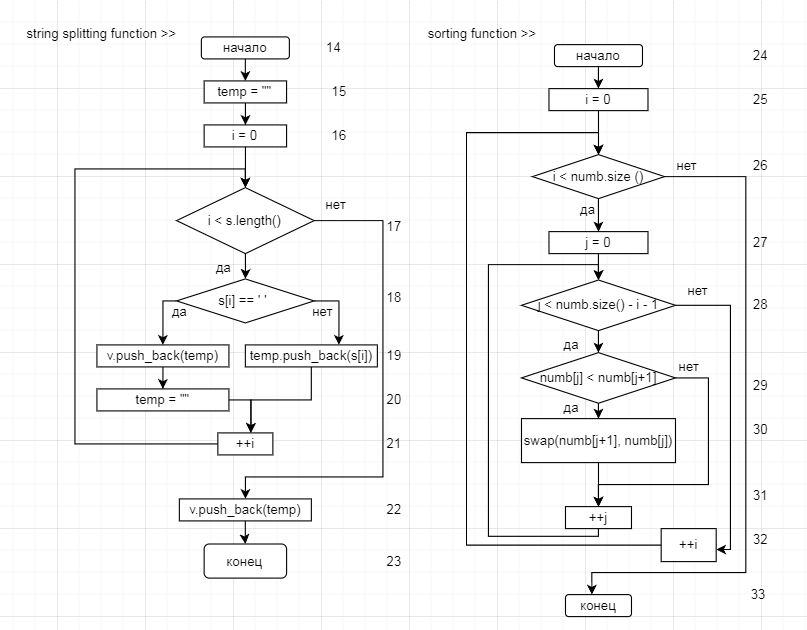
Задан текст. Определить сколько раз встречается в тексте каждое слово.

1. Таблица имён:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Рабочие переменные | | | | | |
| i | integer | | | Счетчик | |
| j | integer | | | Счетчик | |
| temp | string | | | Временная переменная | |
| s | string | | | Временная переменная | |
| numb | vector | | | Временная переменная | |
| Исходные данные | | | | | |
| in | | string | | Текст | |
| Результат | | | | | |
| strings | | | vector | | Слово |
| count | | | integer | | Счетчик |

1. Блок-схема:



1. Отладочный пример:
2. Начало
3. Вывод ‘enter text’
4. Ввод текста (**biba boba biba**)
5. Вызов внешней функции splitstring
6. Вызов внешней функции fsort
7. Count = 1
8. i = 0
9. Проверка условия (i < strings.size) (0 < 3 => верно)
10. Проверка условия (strings[0] = strings[1]) (неверно)
11. Вывод strings[1], count
12. count = 1
13. i++

8. Проверка условия (i < strings.size) (1 < 3 => верно)

9.Проверка условия (strings[1] = strings[2]) (верно)

10.Count ++ (**2**)

11.–

12.i++

………….

1. Конец программы

Внешние функции

splitstring

1. Начало внешней функции
2. Temp = “”
3. i = 0
4. Проверка условия (i<s.length()) (0 < 14 => верно)
5. Проверка условия (s[i] == ‘ ‘) (b != ‘ ‘ => неверно)
6. S[0] = “”
7. –
8. ++i

………

1. Возврат преобразованного вектора
2. Завершение функции splitstring, возврат к программе

fsort

1. Начало функции fsort
2. i = 0
3. Проверка условия (i<numb.size()) (0<3 => верно)
4. j = 0
5. Проверка условия (j<numb.size() – i - 1) (0 < 2 => верно)
6. Проверка условия (numb[j] < numb[j+1]) (boba < biba => неверно)
7. -
8. ++j

28. проверка условия

………..

1. ++i

26. Проверка условия

………..

1. Завершение функции splitstring, возврат к программе
2. Код программы:

#include <iostream>//подключение библиотек

#include <string>

#include <vector>

using namespace std;

void fsort(vector <string>& numb) { //сортировка

for (int i = 0; i < numb.size(); ++i) {

for (int j = 0; j < numb.size() - i - 1; ++j) {

if (numb[j] < numb[j + 1])

swap(numb[j + 1], numb[j]);

}

}

}

void splitString(string s, vector<string>& v) {//преобразование текста

string temp = "";//временная переменная

for (int i = 0; i < s.length(); ++i) {

if (s[i] == ' ') {//антипробел

v.push\_back(temp);

temp = "";

}

else {

temp.push\_back(s[i]);

}

}

v.push\_back(temp);

}

int main()//основная функция

{

vector<string> strings; //объявление переменных

string in;

cout << "enter text : \n";//вывод

getline(cin, in); //ввод

SplitString(in, strings); //вызов функций

fsort(strings);

int count = 1;

for (int i = 0; i < strings.size(); i++)

{

if (strings[i] == strings[i + 1]) //проверка условия

{

count++;

}

else

{

cout << strings[i] << ":" << count << " ";//вывод

count = 1;

}

}

return 0;

}

1. Результат выполнения работы программы:



1. Вывод:

В ходе выполнения работы были изучены базовые алгоритмы и функции языка высокого уровня C++.

Был произведен анализ их работы, на основании которого была составлена блок-схема.

На контрольных примерах было выяснено, что все написанные функции работают корректно и в коде отсутствуют ошибки.

Был оформлен комплект документации на программный код.