

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



## ОТЧЁТ

О выполнении лабораторной работы № 4

"Работа со строками"

**Студент:** Ким В.А.

**Группа:** Б22-703

**Преподаватель:** Овчаренко Е.С.

Москва - 2022

## **1. Формулировка индивидуального задания**

Вариант № 9.

### **Задание**

Удалить из строки все группы повторяющихся подряд одинаковых слов.

## **2. Описание использованных типов данных**

В данной работе использовались тип данных «int» для работы с целыми числами, char\* для работы со строками

## **3. Описание использованного алгоритма**

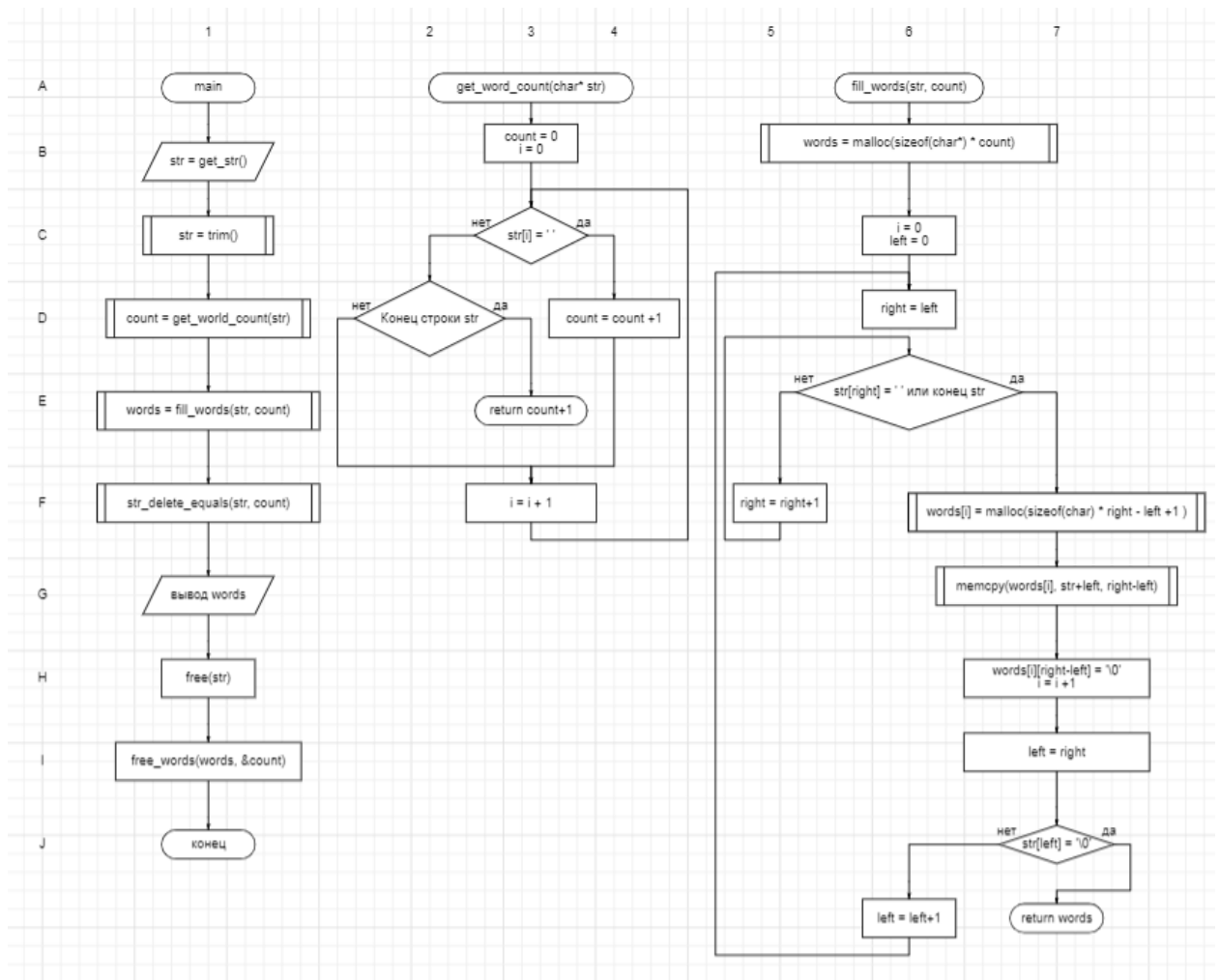


Рис. 1: Блок-схема алгоритма работы функции `main`, `get_word_count`, `fill_word`

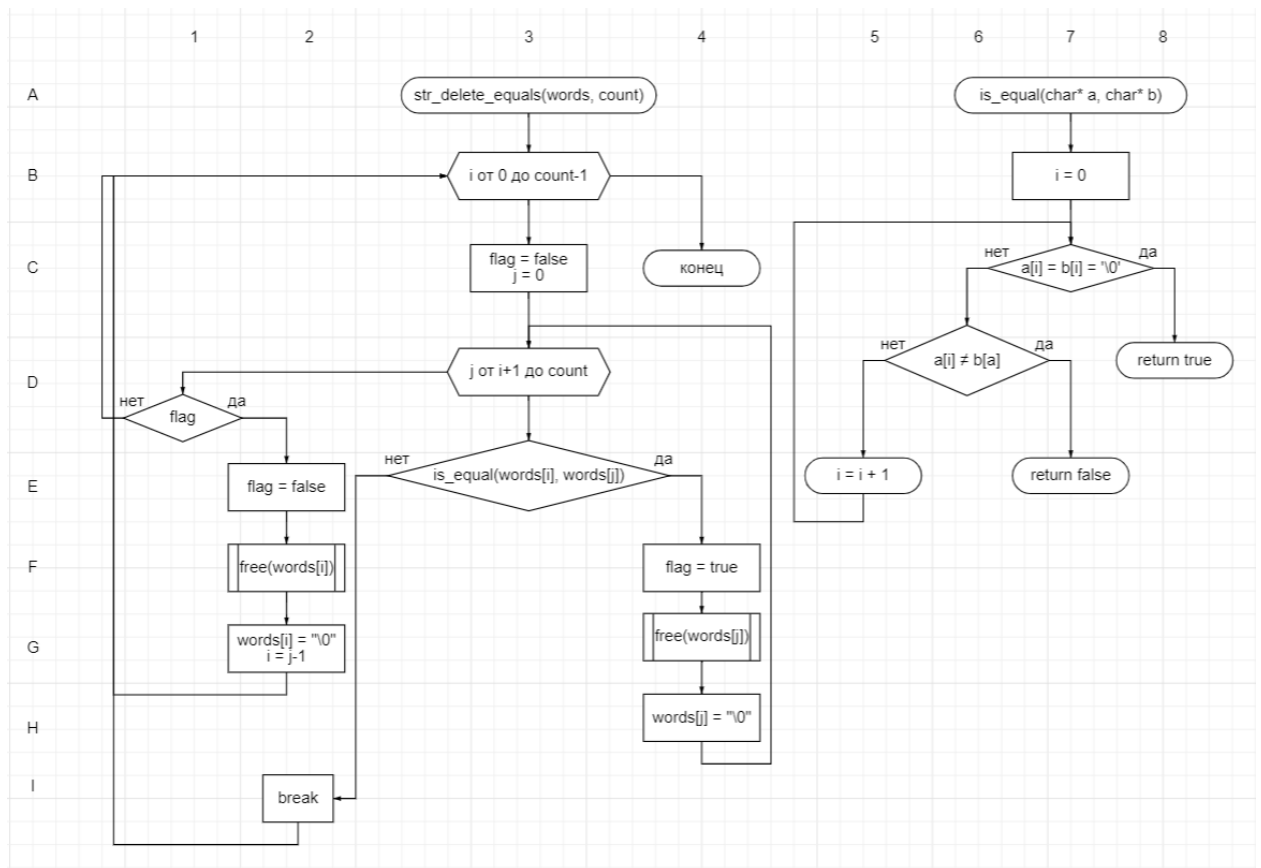


Рис. 2: Блок-схема алгоритма работы функции str\_delete\_equals и is\_equal

## 4. Исходные коды разработанных программ

Листинг 1: Исходные код программы файл: lab\_04.c

```

1  #include <locale.h>
2  #include <stdio.h>
3  #include <stdbool.h>
4  #include <malloc.h>
5  #include <readline/readline.h>
6  #include <time.h>
7
8  #include "./lib_scanf.h"
9
10 unsigned get_word_count(const char* const str) {
11     unsigned count = 0;
12     for (unsigned i = 0;; ++i) {
13         if (IS_SPACE(str[i])) ++count;
14         else if (str[i] == '\\0') return ++count;
15     }
16 }
17
18 char** fill_words(const unsigned count, const char* const str) {
19     char** words = malloc(sizeof(char*) * count);
20     if (words == NULL) throw(null_ptr);
21

```

```

22     unsigned i = 0;
23
24     for (unsigned left = 0;; ++left) {
25         for (unsigned right = left;; ++right) {
26             if (IS_SPACE(str[right]) || str[right] == '\0') {
27                 words[i] = malloc(sizeof(char) * (right - left + 1));
28                 if (words[i] == NULL) throw(null_ptr);
29
30                 memcpy(words[i], str + left, right - left);
31                 words[i++][right - left] = '\0';
32                 left = right;
33                 break;
34             }
35         }
36
37         if (str[left] == '\0') break;
38     }
39
40     return words;
41 }
42
43 static bool is_equal(const char* const a, const char* const b) {
44     if (
45         str_len(a) != str_len(b)
46     ) return false;
47
48     for (unsigned i = 0; a[i] != '\0'; ++i)
49         if (a[i] != b[i]) return false;
50
51     return true;
52 }
53
54 void str_delete_equals(char** const words, const unsigned count) {
55     for (unsigned i = 0; i < count - 1; ++i) {
56         bool flag = false;
57
58         unsigned j;
59         for (j = i + 1; j < count; ++j) {
60             if (is_equal(words[i], words[j])) {
61                 flag = true;
62
63                 free(words[j]);
64                 words[j] = malloc(sizeof(char));
65                 if (words[j] == NULL) throw(null_ptr);
66
67                 words[j][0] = '\0';
68             }
69             else break;
70         }
71         if (flag) {
72             free(words[i]);
73
74             words[i] = malloc(sizeof(char));
75             if (words[i] == NULL) throw(null_ptr);
76             words[i][0] = '\0';
77
78             flag = false;

```

```

79     i = j - 1;
80 }
81 }
82 }
83
84 void free_words(char** words, unsigned* count) {
85     for (unsigned i = 0; i < *count; ++i)
86         free(words[i]);
87
88     free(words);
89     *count = 0;
90 }
91
92 int main() {
93     setlocale(LC_ALL, "rus");
94
95     char* str = trim(
96         get_str("Введите строку: ")
97     );
98
99     unsigned count = get_word_count(str);
100
101     char** words = fill_words(count, str);
102     str_delete_equals(words, count);
103
104     for (int i = 0; i < count; ++i)
105         if (words[i][0] != '\0') printf("%s ", words[i]);
106
107     free(str);
108     free_words(words, &count);
109
110     return 0;
111 }
112

```

## 5. Описание текстовых примеров

### Программа №1

Значения массива	Ожидаемые значения	Полученное значение В
Hello world world	Hello	Hello
__lol_lol_ok__	Ok	Ok

## 6. Скриншоты

```
[kim.va@unix:~/labs/lab_04]$ ./lol
Введите строку: "Hello world world
"
Time of readline: 0.00019
"Hello "[kim.va@unix:~/labs/lab_04]$ ./lol
Введите строку: " lol lol ok
"
Time of readline: 0.00015
"ok "[kim.va@unix:~/labs/lab_04]$
```

## 7. Выводы

В ходе выполнения данной работы на примере программы, выполняющей удаление повторяющихся слов, мы научились записывать информацию из потока ввода, работать со строками.