

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



ОТЧЕТ

**О выполнении лабораторной работы №4
«Работа со строками»**

Студент: Титов Д. И.

Группа: Б22-505

Преподаватель: Вавренюк А. Б.

Москва — 2022

1. Формулировка индивидуального задания

Вариант №2. Удалить все согласные буквы из строки.

2. Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных `int`, `char`, указатели на данные типы данных, так же `double`, `clock_t`, `const char` и указатель на `const char`, также использовались типы данных `void*`, `const void*` и `size_t`

3. Описание использованного алгоритма

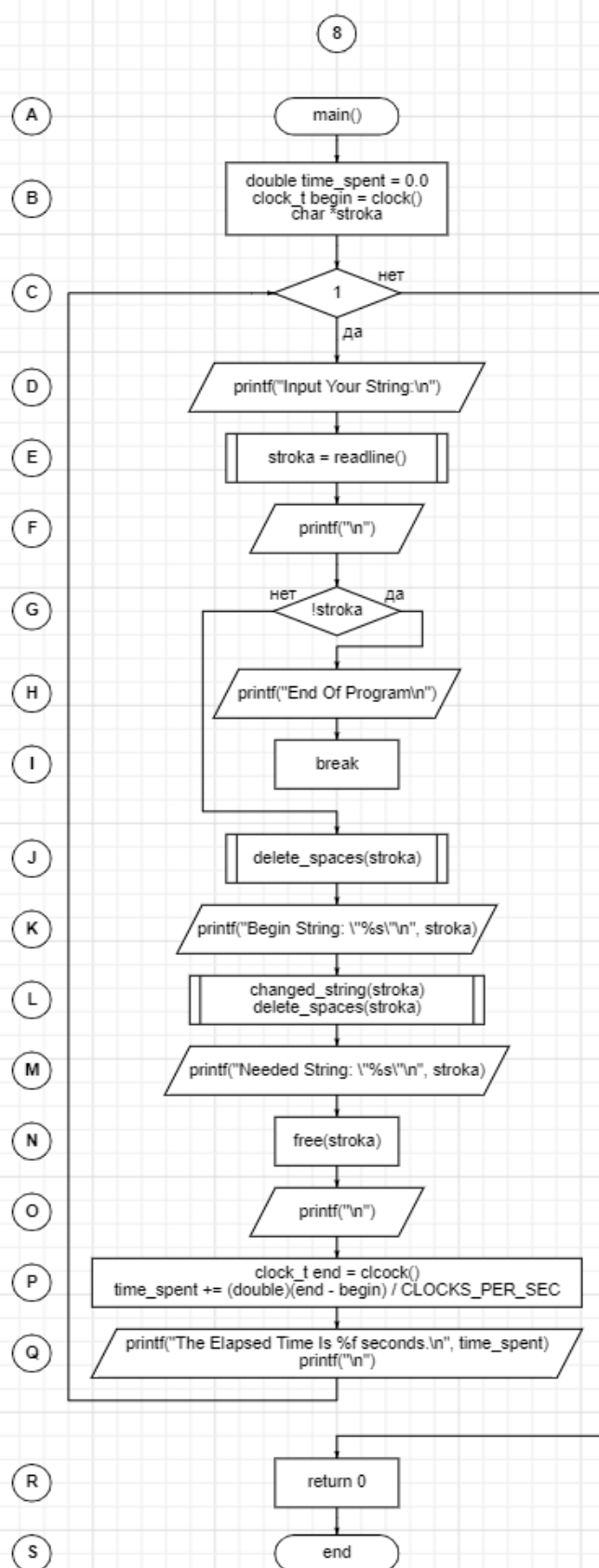


Рис. 1: Блоксхема алгоритма работы функции `main()`

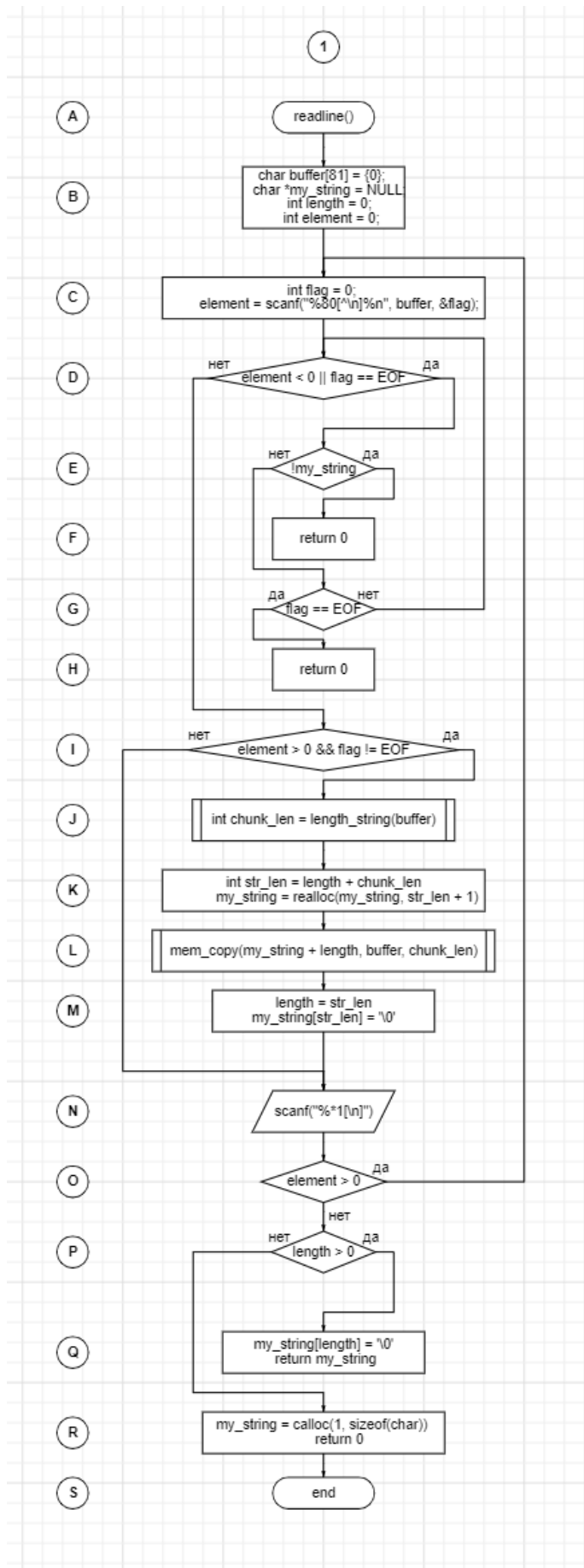


Рис. 2: Блоксхема алгоритма работы функции `readline()`

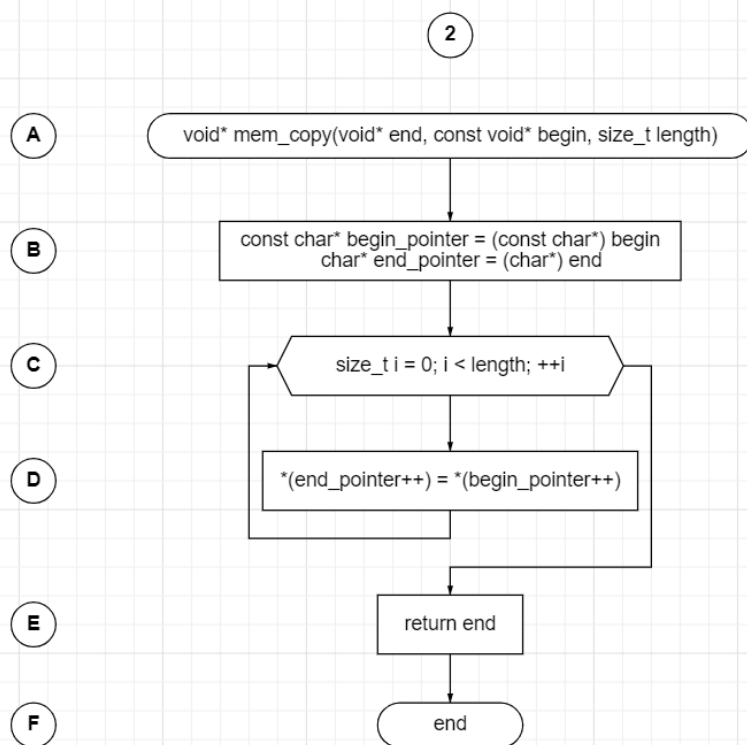


Рис. 3: Блоксхема алгоритма работы функции `mem_copy()`

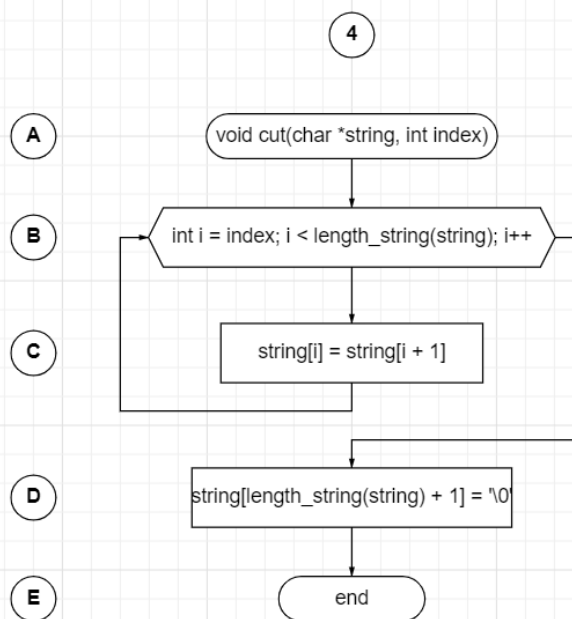


Рис. 4: Блоксхема алгоритма работы функции `cut()`

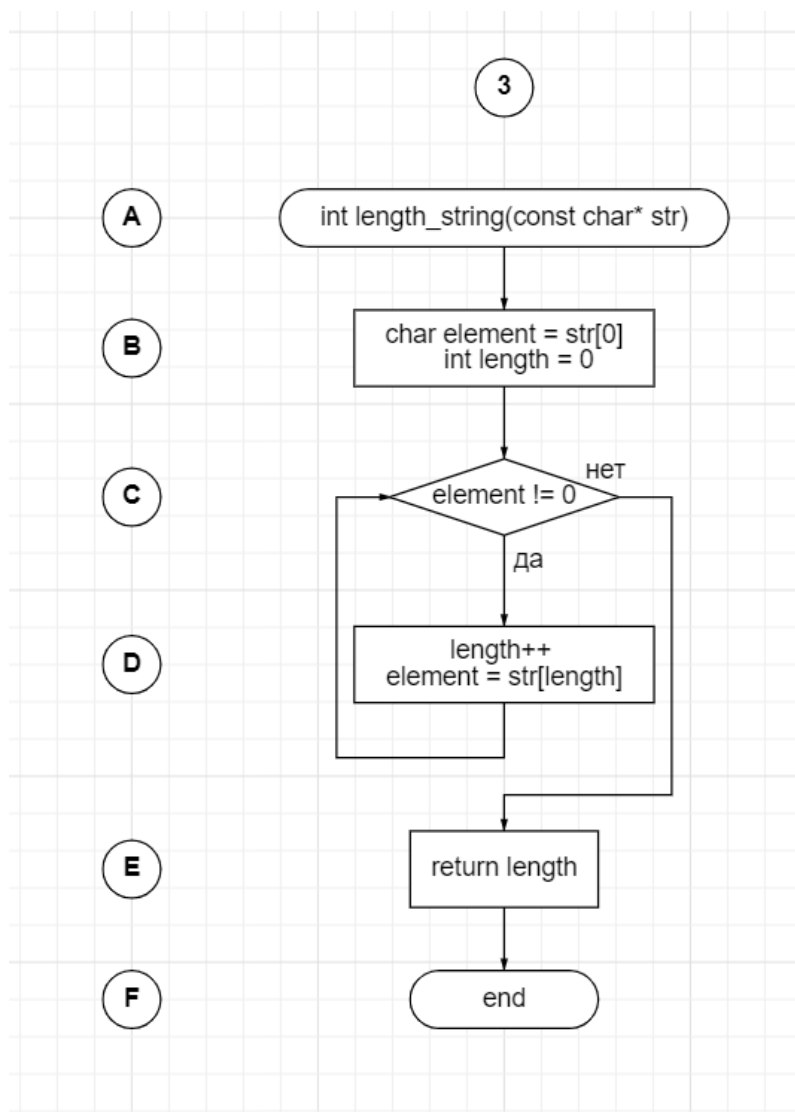


Рис. 5: Блоксхема алгоритма работы функции `length_string()`

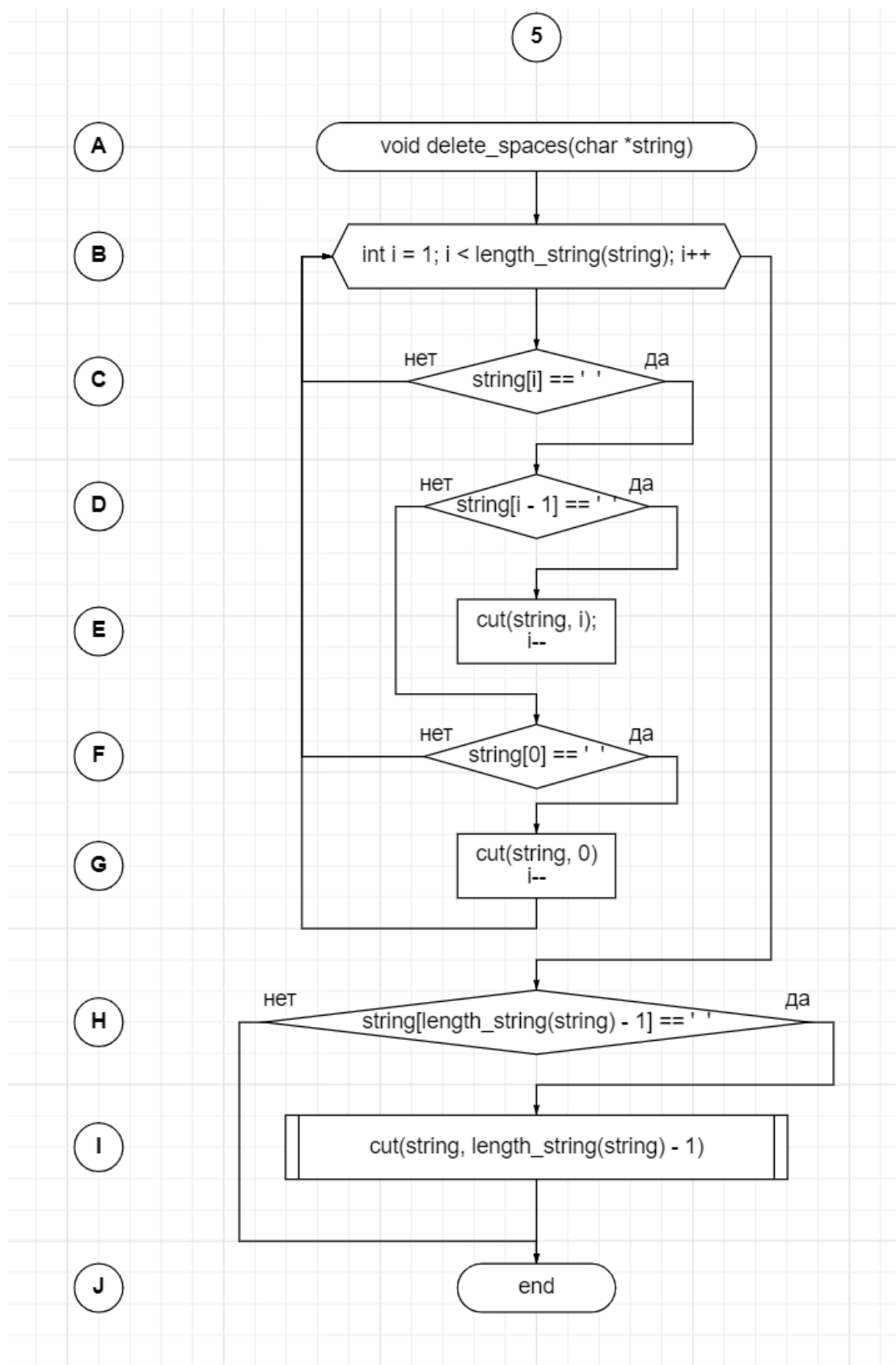


Рис. 6: Блоксхема алгоритма работы функции `delete_spaces()`

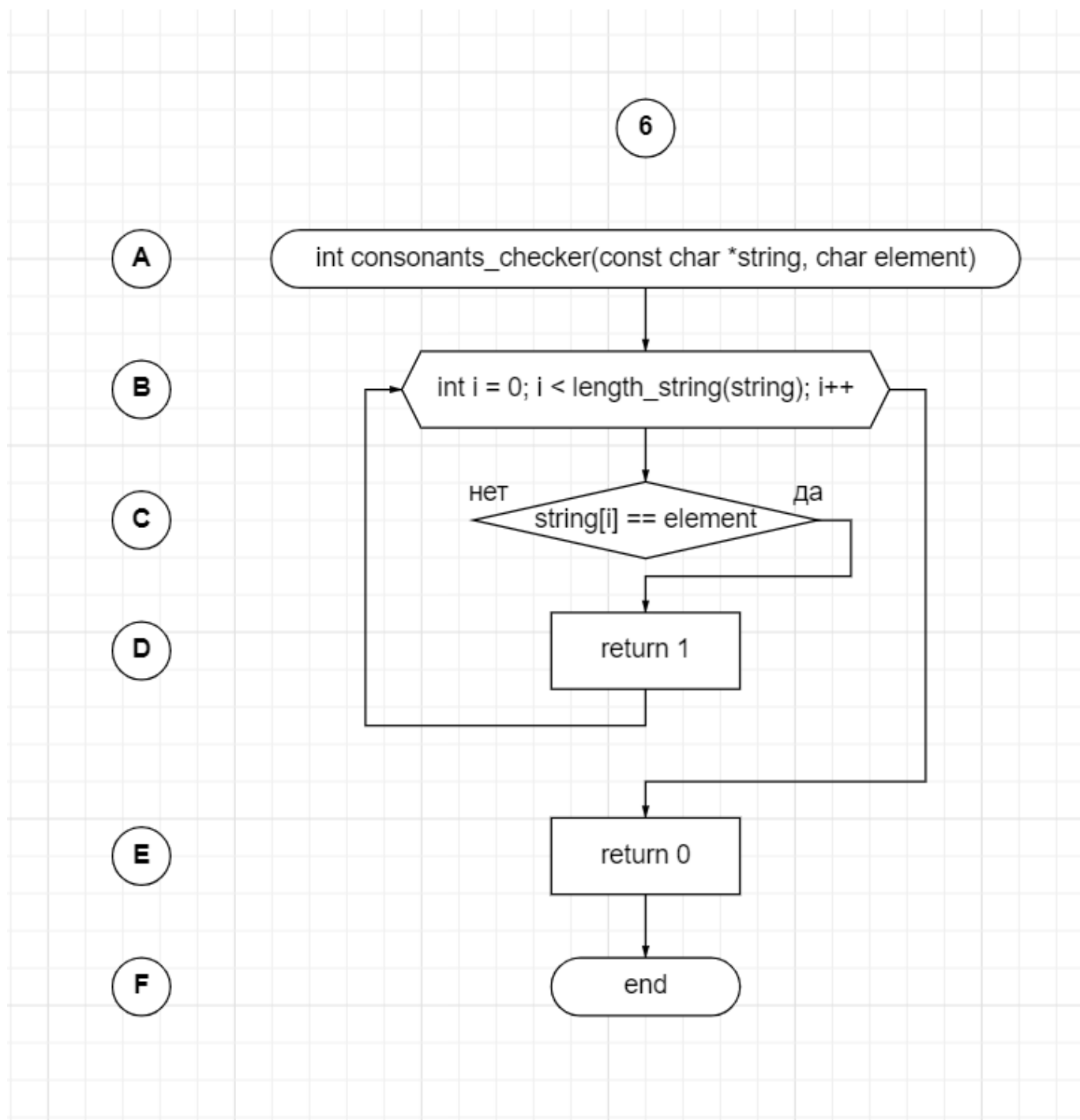


Рис. 7: Блоксхема алгоритма работы функции `consonants_checker()`

Рис. 1: Блоксхема алгоритма работы функции `main()`

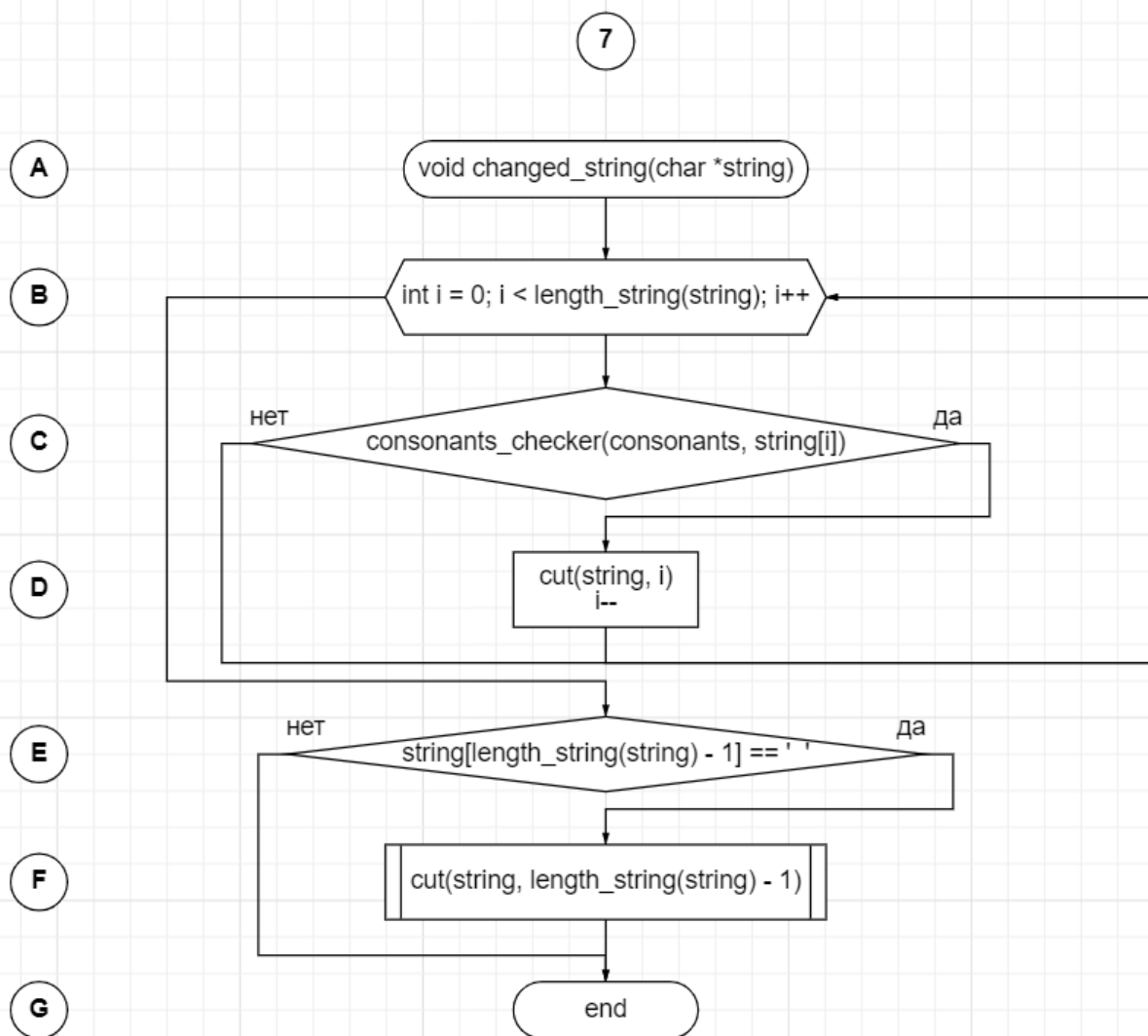


Рис. 8: Блоксхема алгоритма работы функции changed_string()

4. Исходные коды разработанных программ

Листинг 1: Исходные коды программы (файл: mainmyreadline.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "myreadline.h"
#include "mystring.h"
#include "taskfunctions.h"

int main() {
    double time_spent = 0.0;
    clock_t begin = clock();
    char *stroka;
    while (1) {
        printf("Input Your String:\n");
        stroka = readline();
        printf("\n");
        if (!stroka) {
            printf("End Of Program.\n");
            break;
        }
        delete_spaces(stroka);
        printf("Your String: \"%s\"\n", stroka);
        changed_string(stroka);
        delete_spaces(stroka);
        printf("Needed String: \"%s\"\n", stroka);
        free(stroka);
        printf("\n");
        clock_t end = clock();
        time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
        printf("The Elapsed Time Is %f Seconds.\n", time_spent);
        printf("\n");
    }
    return 0;
}
```

Листинг 2: Исходные коды программы (файл: mainreadline.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <readline/readline.h>
#include "taskfunctions.h"

int main() {
    double time_spent = 0.0;
    clock_t begin = clock();
    char *stroka, buffer[81];
    while (1) {
        printf("Input Your String:\n");
        stroka = readline(buffer);
        printf("\n");
        if (!stroka) {
            printf("End Of Program.\n");
            break;
        }
        delete_spaces(stroka);
        printf("Your String: \"%s\"\n", stroka);
        changed_string(stroka);
        delete_spaces(stroka);
        printf("Needed String: \"%s\"\n", stroka);
        free(stroka);
        printf("\n");
        clock_t end = clock();
        time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
        printf("The Elapsed Time Is %f Seconds.\n", time_spent);
        printf("\n");
    }
    return 0;
}
```

Листинг 3 : Исходные коды программы (файл: myreadline.c)

```
#include <stdio.h>
#include <stdlib.h>
#include "myreadline.h"
#include "mystring.h"

char* readline()
{
    char buffer[81] = {0};
    char *my_string = NULL;
    int length = 0;
    int element = 0;
    do {
        element = scanf("%80[^\n]", buffer);
        if (element < 0 || element == EOF) {
            if (!my_string) {
                return 0;
            } else if (element == EOF) {
                return 0;
            }
        } else if (element > 0 && element != EOF) {
            int chunk_len = length_string(buffer);
            int str_len = length + chunk_len;
            my_string = realloc(my_string, str_len + 1);
            mem_copy(my_string + length, buffer, chunk_len);
            length = str_len;
            my_string[str_len] = '\0';
        } else {
            scanf("%*1[\n]");
        }
    } while (element > 0);
    if (length > 0) {
        my_string[length] = '\0';
        return my_string;
    } else {
        my_string = calloc(1, sizeof(char));
        return 0;
    }
}
```

Листинг 4: Исходные коды программы (файл: mystring.c)

```
#include <stdio.h>
#include "mystring.h"

void* mem_copy(void* end, const void* begin, size_t length) {
    const char* begin_pointer = (const char*) begin;
    char* end_pointer = (char*) end;
    for (size_t i = 0; i < length; ++i) {
        *(end_pointer++) = *(begin_pointer++);
    }
    return end;
}

int length_string(const char* str) {
    char element = str[0];
    int length = 0;
    while (element != 0) {
        length++;
        element = str[length];
    }
    return length;
}
```

Листинг 5 : Исходные коды программы (файл: taskfunctiononmystring.c)

```
#include <stdio.h>
#include "taskfunctions.h"
#include "mystring.h"

#define consonants "BCDFGHJKLMNPQRSTVWXZbcdfghjklmnpqrstvwxyz"

void cut(char *string, int index) {
    for (int i = index; i < length_string(string); i++) {
        string[i] = string[i + 1];
    }
    string[length_string(string) + 1] = '\0';
}

void delete_spaces(char *string) {
    for (int i = 1; i < length_string(string); i++) {
        if (string[i] == ' ') {
            if (string[i - 1] == ' ') {
                cut(string, i);
                i--;
            } else if (string[0] == ' ') {
                cut(string, 0);
                i--;
            }
        }
    }
    if (string[length_string(string) - 1] == ' ') {
        cut(string, length_string(string) - 1);
    }
}

int consonants_checker(const char *string, char element) {
    for (int i = 0; i < length_string(string); i++) {
        if (string[i] == element) {
            return 1;
        }
    }
    return 0;
}

void changed_string(char *string) {
    for (int i = 0; i < length_string(string); i++) {
        if (consonants_checker(consonants, string[i])) {
            cut(string, i);
            i--;
        }
    }
}
```

Листинг 6 : Исходные коды программы (файл: taskfunctionsnotonmystring.c)

```
#include <stdio.h>
#include "taskfunctions.h"
#include <string.h>

#define consonants "BCDFGHJKLMNPQRSTVWXZbcdfghjklmnpqrstvwxyz"

void cut(char *string, int index) {
    for (int i = index; i < strlen(string); i++) {
        string[i] = string[i + 1];
    }
    string[strlen(string) + 1] = '\0';
}

void delete_spaces(char *string) {
    for (int i = 1; i < strlen(string); i++) {
        if (string[i] == ' ') {
            if (string[i - 1] == ' ') {
                cut(string, i);
                i--;
            } else if (string[0] == ' ') {
                cut(string, 0);
                i--;
            }
        }
    }
    if (string[strlen(string) - 1] == ' ') {
        cut(string, strlen(string) - 1);
    }
}

int consonants_checker(const char *string, char element) {
    for (int i = 0; i < strlen(string); i++) {
        if (string[i] == element) {
            return 1;
        }
    }
    return 0;
}

void changed_string(char *string) {
    for (int i = 0; i < strlen(string); i++) {
        if (consonants_checker(consonants, string[i])) {
            cut(string, i);
            i--;
        }
    }
}
```

5. Описание тестовых примеров

Таблица 1: Тестовые примеры programwithminereadline

Вводимы Данные	Ожидаемый Ответ	Полученный Ответ
<u>;lqjahe45 ;oiu12oi4 upo</u>	Your String: <u>;lqjahe45 ;oiu12oi4 upo</u> Needed String: <u>;ae45 ;oiu12oi4 uo</u> The Elapsed Time Is ? Seconds.	Your String: <u>;lqjahe45 ;oiu12oi4 upo</u> Needed String: <u>;ae45 ;oiu12oi4 uo</u> The Elapsed Time Is 0.000383 Seconds.
<u>lewjth 1234 'las q234 d</u>	Your String: <u>lewjth 1234 'las q234 d</u> Needed String: e 1234 'as 234 The Elapsed Time Is ? Seconds.	Your String: Needed String: The Elapsed Time Is 0.001065 Seconds.

Таблица 2: Тестовые примеры programwithnotminereadline

Вводимы Данные	Ожидаемый Ответ	Полученный Ответ
<u>;lqjahe45 ;oiu12oi4 upo</u>	Your String: <u>;lqjahe45 ;oiu12oi4 upo</u> Needed String: <u>;ae45 ;oiu12oi4 uo</u> The Elapsed Time Is ? Seconds.	Your String: <u>;lqjahe45 ;oiu12oi4 upo</u> Needed String: <u>;ae45 ;oiu12oi4 uo</u> The Elapsed Time Is 0.005205 Seconds.
<u>lewjth 1234 'las q234 d</u>	Your String: <u>lewjth 1234 'las q234 d</u> Needed String: e 1234 'as 234 The Elapsed Time Is ? Seconds.	Your String: Needed String: The Elapsed Time Is 0.012718 Seconds.

6. Скриншоты

```
[titov.di@unix:~/Lab4]$ ./programwithnotminereadline
Input Your String:
;lqjahe45 ;oiu12oi4 upo

Your String: ";lqjahe45 ;oiu12oi4 upo"
Needed String: ";ae45 ;oiu12oi4 uo"

The Elapsed Time Is 0.005205 Seconds.

Input Your String:
lewjth 1234 ';las q234 d

Your String: "lewjth 1234 ';las q234 d"
Needed String: "e 1234 ';a 234"

The Elapsed Time Is 0.012718 Seconds.

Input Your String:

End Of Program.
[titov.di@unix:~/Lab4]$ ./programwithminereadline
Input Your String:
;lqjahe45 ;oiu12oi4 upo

Your String: ";lqjahe45 ;oiu12oi4 upo"
Needed String: ";ae45 ;oiu12oi4 uo"

The Elapsed Time Is 0.000383 Seconds.

Input Your String:
lewjth 1234 ';las q234 d

Your String: "lewjth 1234 ';las q234 d"
Needed String: "e 1234 ';a 234"

The Elapsed Time Is 0.001065 Seconds.

Input Your String:

End Of Program.
```

Рис. 9: Запуск программ, обеих для проверки скорости работы

7. Выводы

В ходе выполнения данной работы на примере программы, выполняющей удаление всех согласных из полученной строки были изучены следующие навыки:

- 1) Работа со строками, написание ряда собственных функций `getline`, а также некоторых функций библиотеки `string.h`
- 2) К тому же скорость работы функции `getline` написанной своими руками оказалась выше, чем скорость встроенной функции в язык Си, что связано с использованными алгоритмами в ходе написания работы