



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»


КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №5

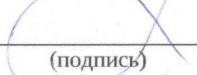
«Обработка строк»

ДИСЦИПЛИНА: «Высокоуровневое программирование»

Выполнил: студент гр. ИУК4-11Б


(подпись) (Суриков Н.С)
(Ф.И.О.)

Проверил:


(подпись) (Пчелинцева Н.И.)
(Ф.И.О.)

Дата сдачи (защиты):

20.11.23

Результаты сдачи (защиты):

- Балльная оценка:

65

- Оценка:

хорошо

Цель работы: изучение приемов работы со строковыми типами данных средствами языка C++.

Задачи:

1. Научиться представлять строку в C++ как массив символов.
2. Изучить основные функции и процедуры обработки строкового типа данных.
3. Изучить типовые алгоритмы решения задач с C-строками.

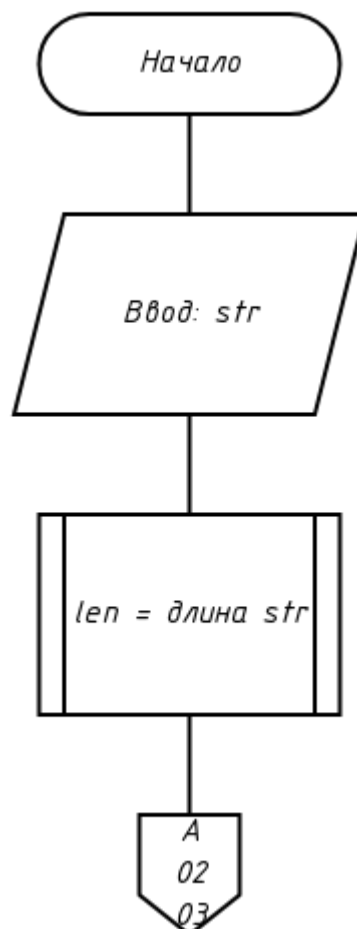
Вариант 20

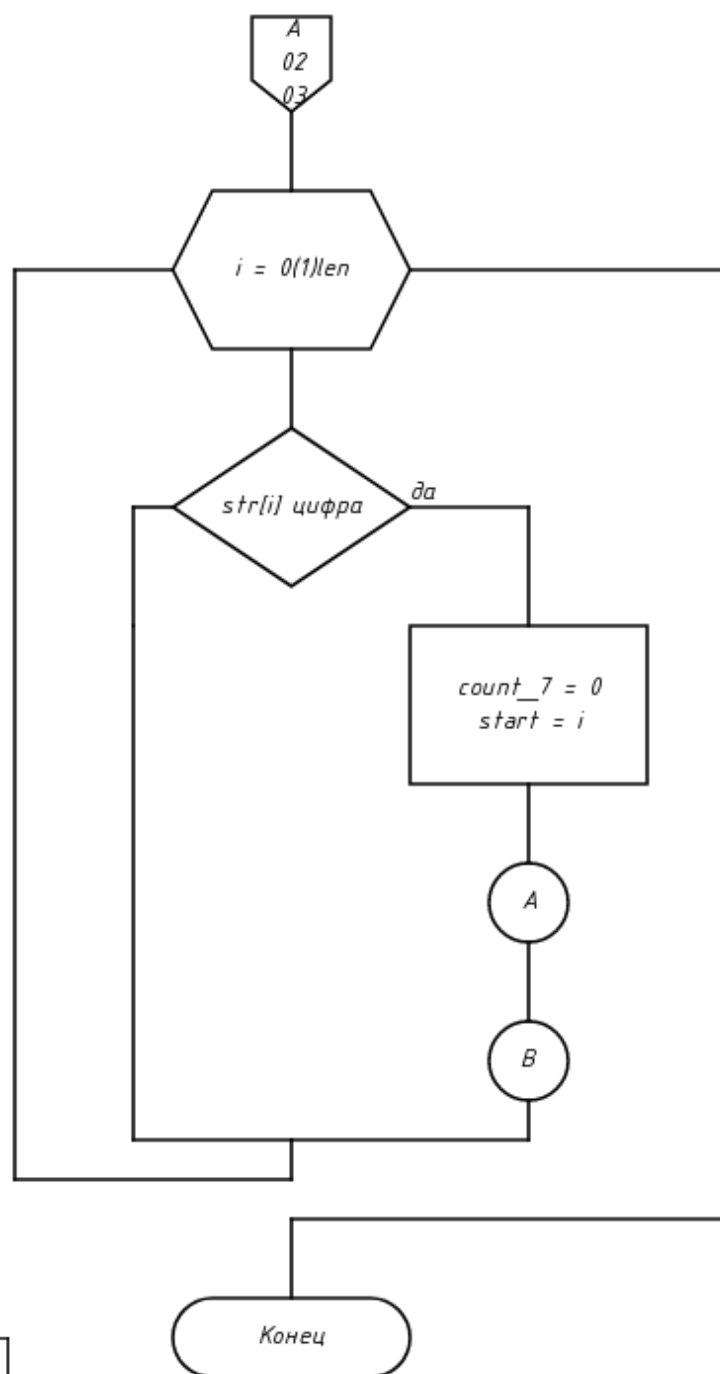
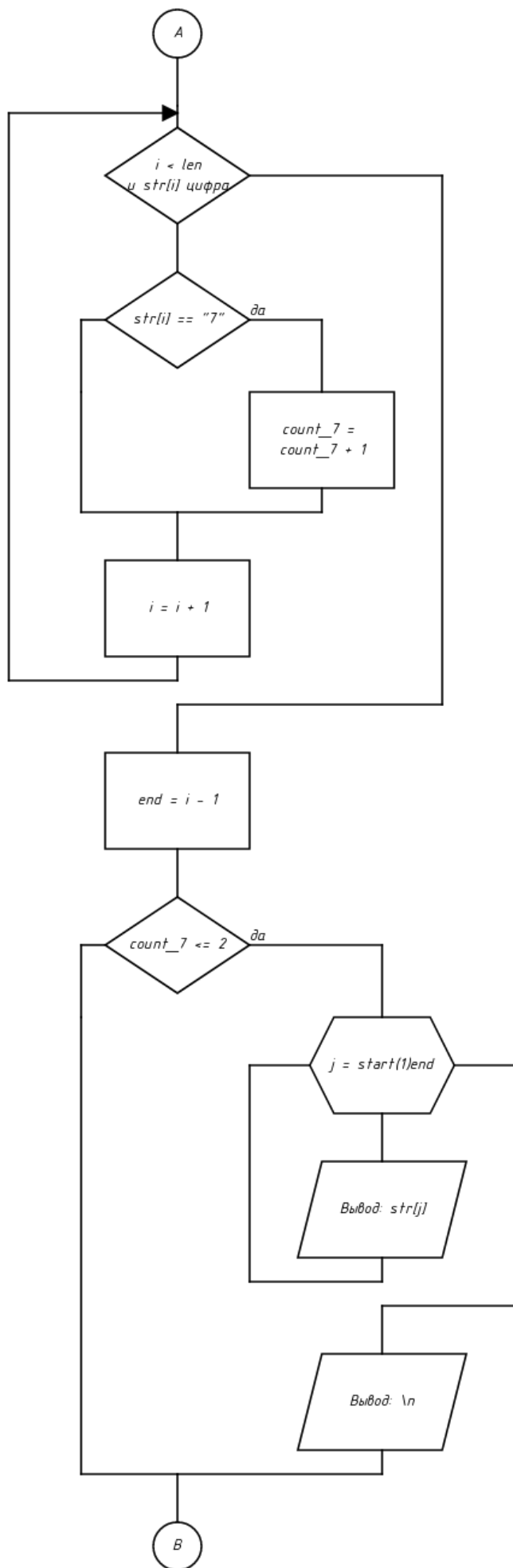
Задание 1:

Формулировка:

Ввести строку, состоящую только из цифр и букв. Распечатать те группы цифр, в которых цифра 7 встречается не более двух раз. (Группа цифр — это последовательность цифр, обрамленная буквами.

Блок схема:





Листинг программы на C++:

```
#include <iostream>
#include <cstring>

using namespace std;

int main() {
    char str[100];
    cin.getline(str, 100);
    int len = strlen(str);

    for (int i = 0; i < len; i++) {
        if (isdigit(str[i])) {
            int count_7 = 0;
            int start = i;
            while (i < len && isdigit(str[i])) {
                if (str[i] == '7') {
                    count_7++;
                }
                i++;
            }
            int end = i - 1;
            if (count_7 <= 2) {
                for (int j = start; j <= end; j++) {
                    cout << str[j];
                }
                cout << endl;
            }
        }
    }

    return 0;
}
```

Результат выполнения:

<i>Входные данные</i>	777	1234567	qwer127dfgegr3 2456677fgr7577 575	q2w2w4we5eet5 yty7hty898yu787 7
<i>Вывод программы</i>		1234567	127 32456677	2 2 4 5 5 7 898

Задание 2:

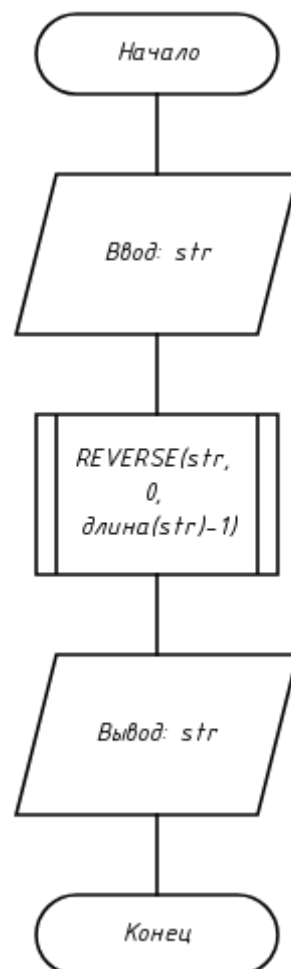
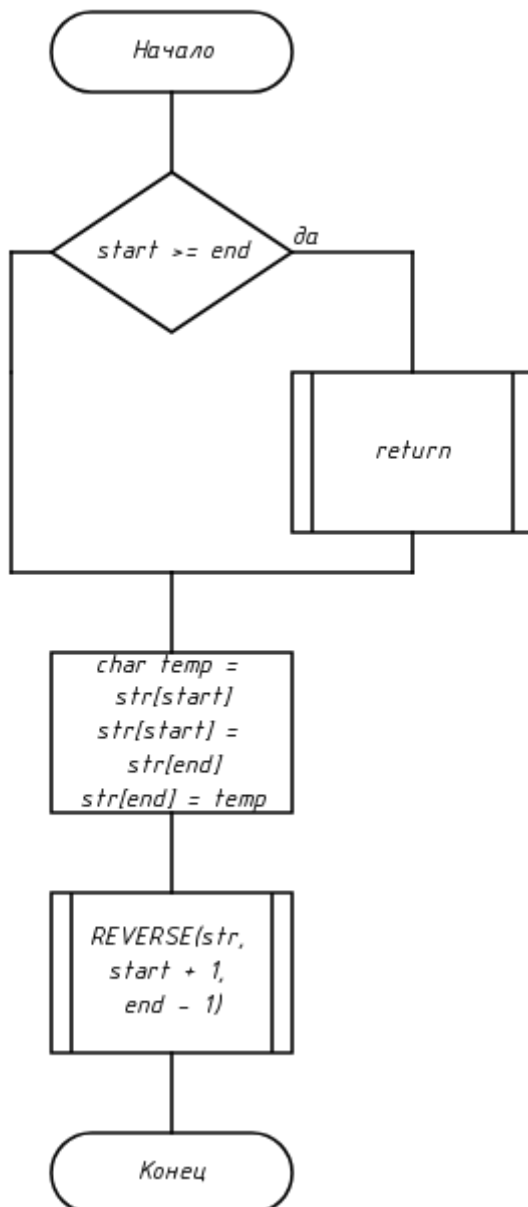
Формулировка:

Написать и протестировать рекурсивную функцию REVERSE(str), которая переворачивает данную строку на том же самом месте

Блок схема:

void REVERSE(char* str, int start, int end)

Основная программа



Листинг программы на C++:

```
#include <iostream>
#include <cstring>

void REVERSE(char* str, int start, int end)
{
    if (start >= end)
        return;

    char temp = str[start];
    str[start] = str[end];
    str[end] = temp;

    REVERSE(str, start + 1, end - 1);
}

int main()
{
    char str[100];
    std::cin.getline(str, 100);

    REVERSE(str, 0, strlen(str) - 1);
    std::cout << str << std::endl;
    return 0;
}
```

Результат выполнения:

<i>Входные данные</i>	<i>qwerty</i>	<i>hello world!</i>	<i>123456789</i>
<i>Вывод программы</i>	<i>ytrewq</i>	<i>!dlrow olleh</i>	<i>987654321</i>

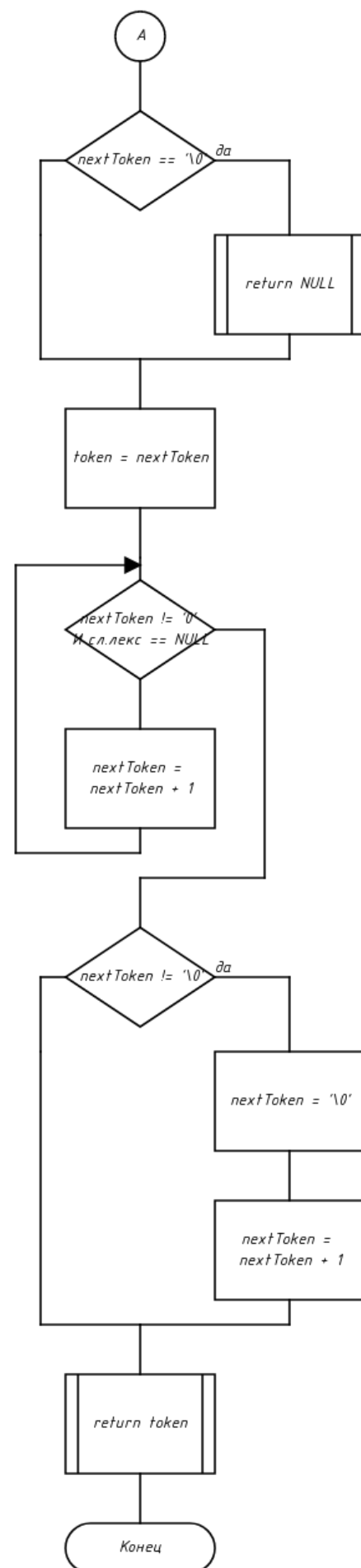
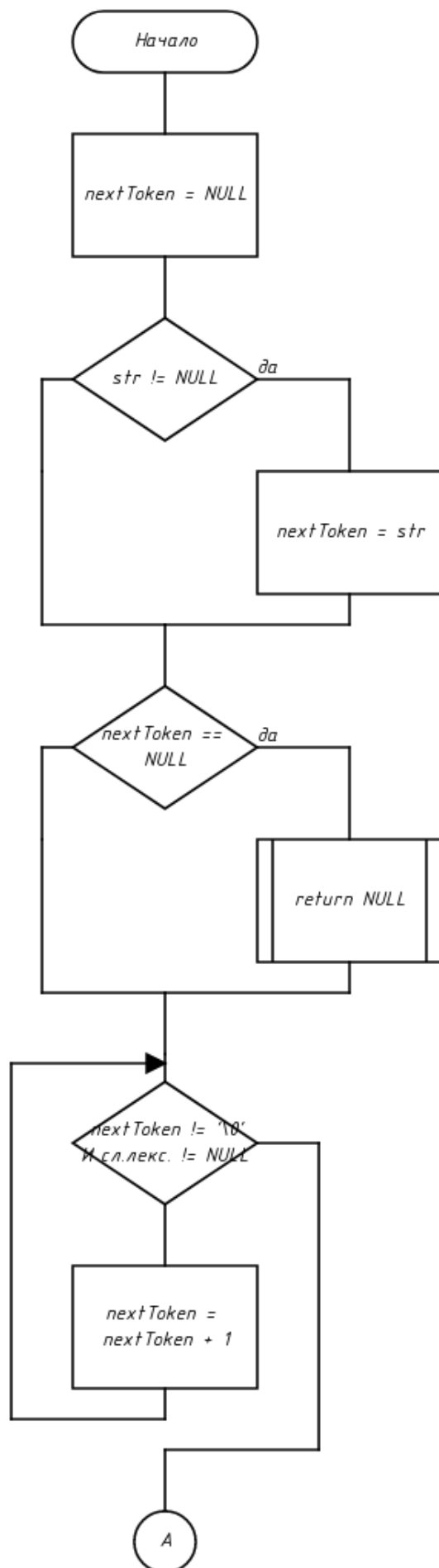
Задание 3:

Формулировка:

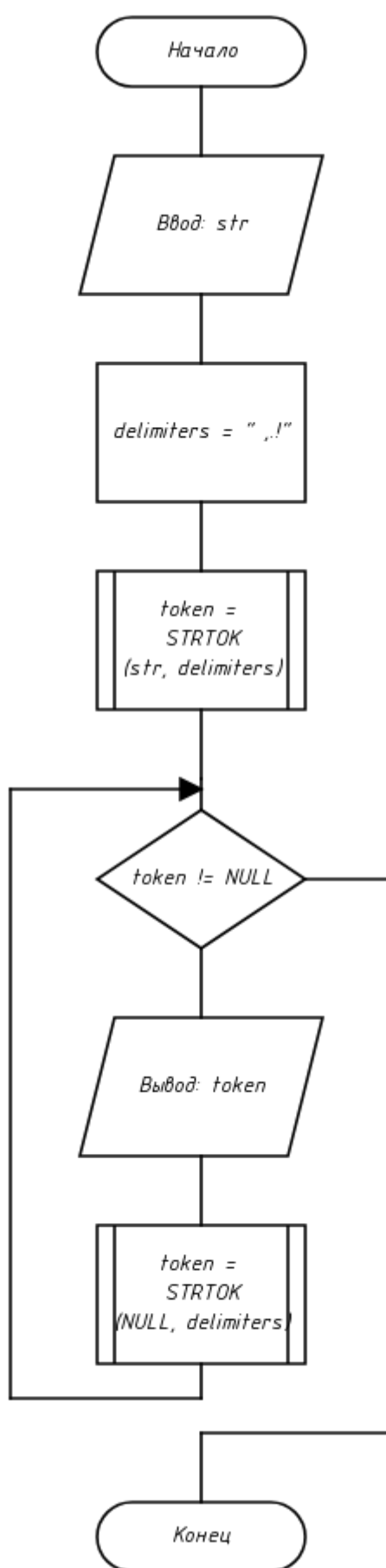
Написать и протестировать аналог функции STRTOK().

Блок схема:

```
char* STRTOK(char* str, const char* delimiters);
```



Основная программа



Листинг программы на C++:

```
#include <cstring>
#include <iostream>

// Функция аналогичная функции strtok()
char* STRTOK(char* str, const char* delimiters) {
    static char* nextToken = nullptr;
    // Если str != nullptr, начинаем обрабатывать новую строку
    if (str != nullptr) {
        nextToken = str;
    }
    // Если nextToken == nullptr, значит больше нет токенов
    if (nextToken == nullptr) {
        return nullptr;
    }
    // Пропускаем ведущие разделители
    while (*nextToken != '\0' && strchr(delimiters,
*nextToken) != nullptr) {
        nextToken++;
    }
    // Если достигнут конец строки, значит больше нет токенов
    if (*nextToken == '\0') {
        return nullptr;
    }
    char* token = nextToken;
    // Находим конец токена
    while (*nextToken != '\0' && strchr(delimiters, *nextToken)
== nullptr) {
        nextToken++;
    }
    // Если достигнут конец строки, то следующий токен - nullptr
    if (*nextToken != '\0') {
        *nextToken = '\0';
        nextToken++;
    }
    return token;
}

int main() {
    char str[100];
    std::cin.getline(str, 100);
    char delimiters[] = " ,.!" ; // Разделители: пробел, запятая,
точка, восклицательный знак
    char* token = STRTOK(str, delimiters);
    // Пока есть токены, выводим их на экран
    while (token != nullptr) {
        std::cout << token << std::endl;
        token = STRTOK(nullptr, delimiters);
    }
    return 0;}
```

Результат выполнения:

Входные данные	123! 345, 567.	123123	123 .!., 122
Вывод программы	123 345 567	123123	123 122

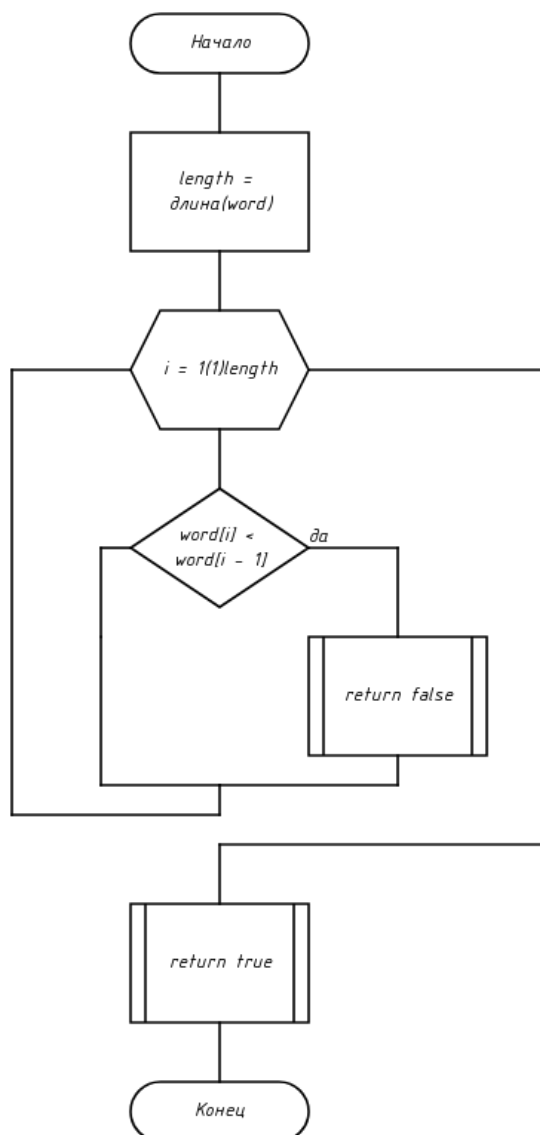
Задание 4:

Формулировка:

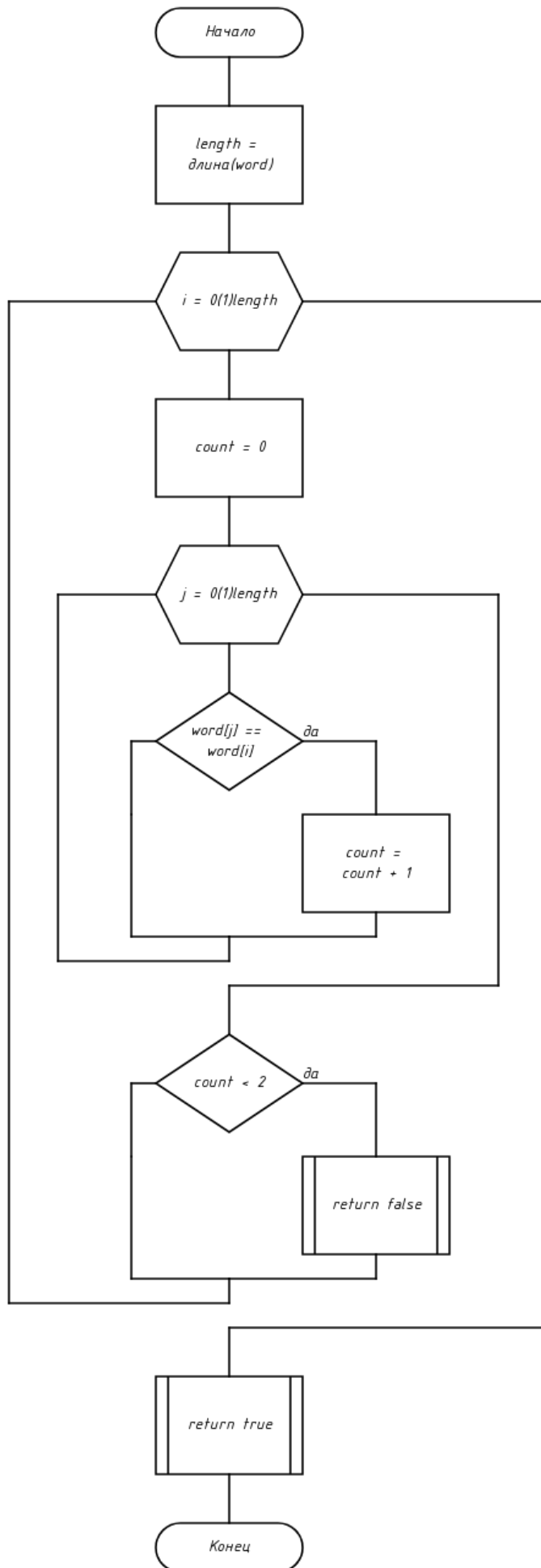
Распечатать те слова, в которых либо буквы упорядочены по алфавиту, либо каждая буква входит в слово не менее двух раз (т.е. слова типа BEER, ABBA).

Блок схема:

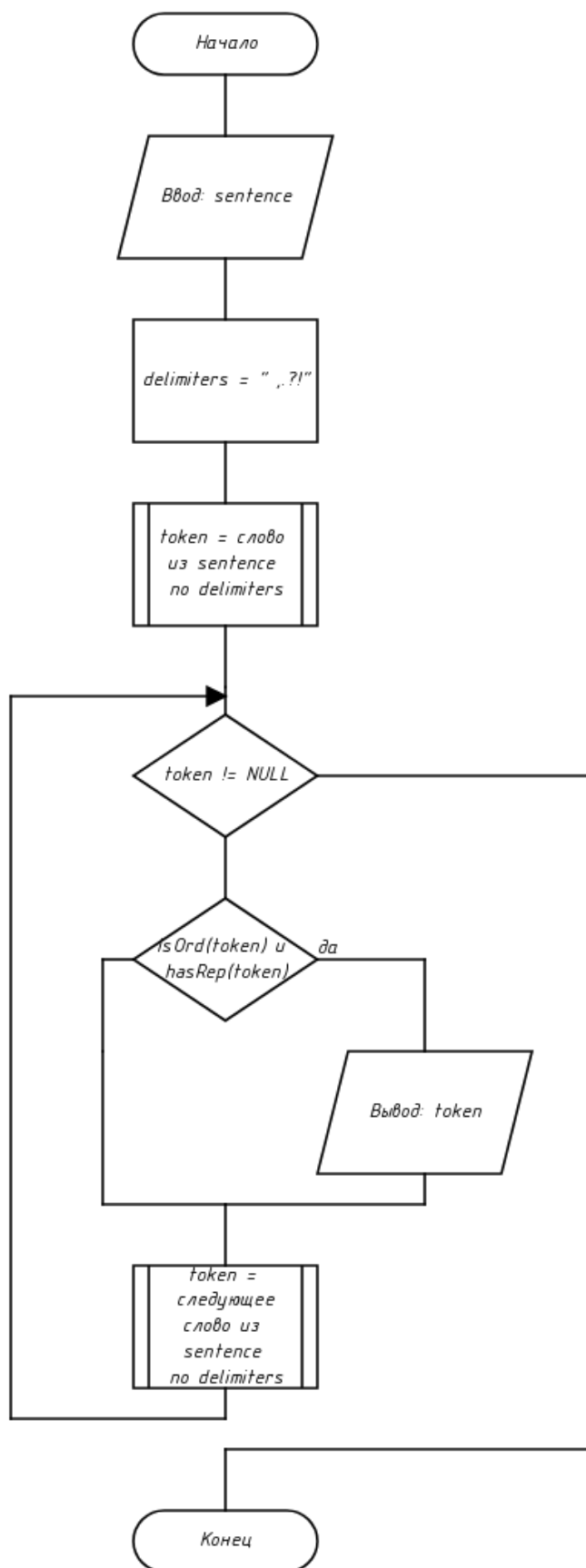
bool isOrd(char *word)



bool hasRep(char *word)



Основная программа



Листинг программы на C++:

```
#include <iostream>
#include <cstring>

bool isOrderedAlphabetically(char *word)
{
    int length = strlen(word);
    for (int i = 1; i < length; i++)
    {
        if (word[i] < word[i - 1])
        {
            return false;
        }
    }
    return true;
}

bool hasRepeatedLetters(char *word)
{
    int length = strlen(word);
    for (int i = 0; i < length; i++)
    {
        int count = 0;
        for (int j = 0; j < length; j++)
        {
            if (word[j] == word[i])
            {
                count++;
            }
        }
        if (count < 2)
        {
            return false;
        }
    }
    return true;
}

int main()
{
    char sentence[30];
    std::cin.getline(sentence, 30);
    const char delimiters[] = " ,.?!" ; // Разделители
    для разделения строки на слова
    char *token = strtok(sentence, delimiters); // Разделение
    строки на слова

    while (token != nullptr)
    {
```

```

        if (isOrderedAlphabetically(token) ||
hasRepeatedLetters(token))
    {
        std::cout << token << std::endl;
    }
    token = strtok(nullptr, delimiters);
}
return 0;
}

```

Результат выполнения:

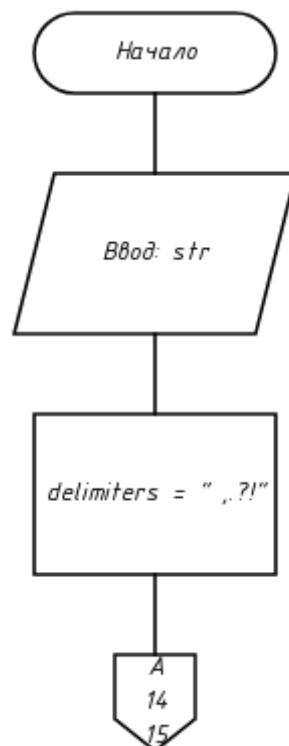
<i>Входные данные</i>	ABBA	abc	ddiigg
<i>Вывод программы</i>	ABBA	abc	ddiigg

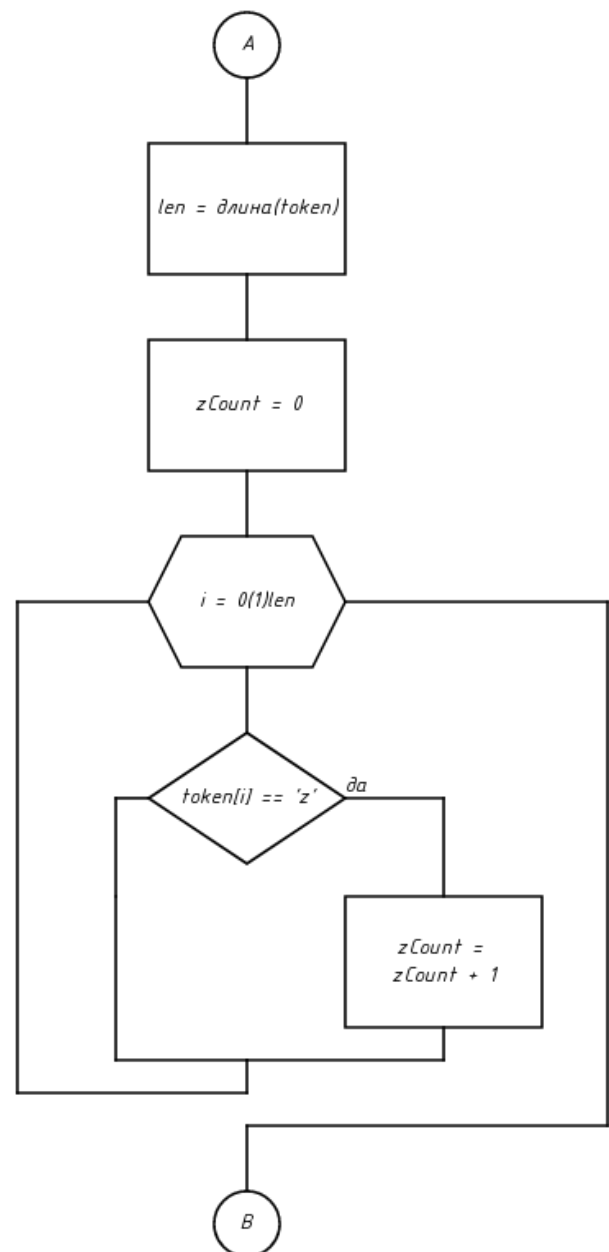
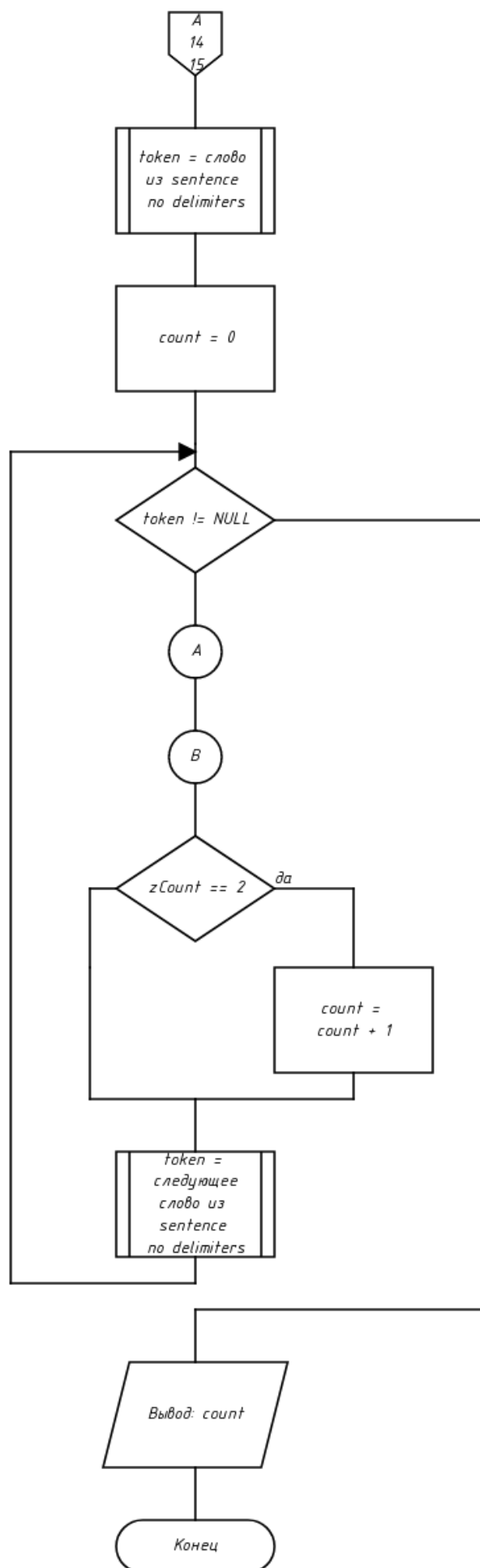
Задание 5:

Формулировка:

Дана последовательность символов латинского алфавита, заканчивающаяся вопросительным знаком. Сосчитать количество слов, в которых ровно две буквы «z». Слова разделяются любым разделителем.

Блок схема:





Листинг программы на C++:

```
#include <iostream>
#include <cstring>

int main() {
    char str[100];
    std::cin.getline(str, 100);
    const char delimiters[] = " \t\n\r,.?!" ;
    char* token = std::strtok(str, delimiters);
    int count = 0;

    while (token != nullptr) {
        int len = std::strlen(token);
        int zCount = 0;

        for (int i = 0; i < len; i++) {
            if (token[i] == 'z') {
                zCount++;
            }
        }

        if (zCount == 2) {
            count++;
            std::cout << token << std::endl;
        }

        token = std::strtok(nullptr, delimiters);
    }

    std::cout << count << std::endl;

    return 0;
}
```

Результат выполнения:

<i>Входные данные</i>	zooz	zz zoo zdz zzzz	zooo zomb ziil
<i>Вывод программы</i>	1	2	0

Вывод

В ходе данной лабораторной работы я изучил приемы работы с строковыми типами данных средствами языка C++, научился представлять строку в C++ как массив символов, изучил основные функции и процедуры обработки строкового типа данных и типовые алгоритмы решения задач с C-строками.

Литература

1. Курс лекций доцента кафедры ФН1-КФ Пчелинцевой Н.И.
2. Зырянов, К. И. Программирование на С++: учебное пособие / К. И. Зырянов, Н. П. Кисленко. — Новосибирск: Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2017. — 129 с. — ISBN 978-5-7795-0817- 9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/85873.html>
3. Фридман, А. Л. Язык программирования С++: учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. — ISBN 978-5-4497-0920-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102076.html>