Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«КУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №3**

«Программирование на языке C++ с использованием строк»

дисциплина «Основы алгоритмизации и программирования»

Выполнил:

студент группы 113 Гришанов Е.А.

Проверил:

старший преподаватель кафедры ПОиАИС Ураева Е.Е.

Курск, 2024

***Цель работы:*** изучить особенности написания программ на языке С++ с использованием строк.

***Задание***

*Задача 1.* Дан текст. Определить частоту появления букв «а» в нем. Частота вычисляется как отношение количества данных символов в тексте к длине всего текста (пробелы учитываются, а символ конца строки не учитывается).

*Задача 2.* Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Для каждой буквы алфавита вывести, в скольких словах этого текста она встречается с учетом регистра и сами слова (например, «a – 1: programming», «b – 0» и т. д).

*Задача 3.* Дан текст, слова в котором разделены одинарным символом пробела, и целое n. Вывести на экран все слова длины n.

***Разработка алгоритма***

*Задача 1*

Входные данные алгоритма решения задачи:

* *inputText* – строка.

Выходные данные алгоритма решения задачи:

* *frequency* – действительное число.

Алгоритм решения задачи представлен на рисунке 1.

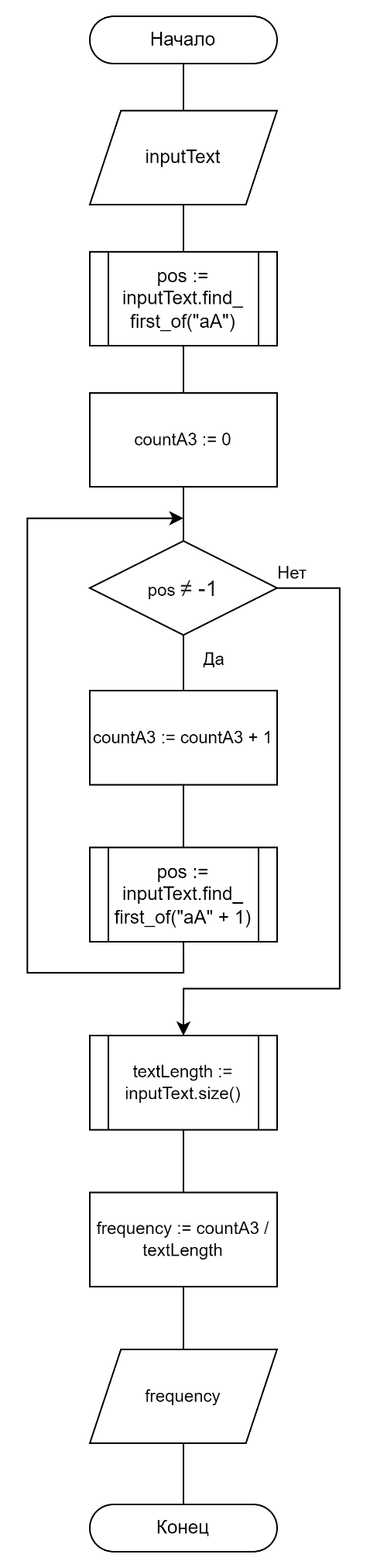


Рисунок 1 – Алгоритм решения задачи 1

*Задача 2*

Входные данные алгоритма решения задачи:

* *text –* строка;

Выходные данные алгоритма решения задачи:

* *letter –* символ;
* *upperLetterCount[] –* массив;
* *lowerLetterCount[] –* массив.

Алгоритм решения задачи представлен на рисунке 2.

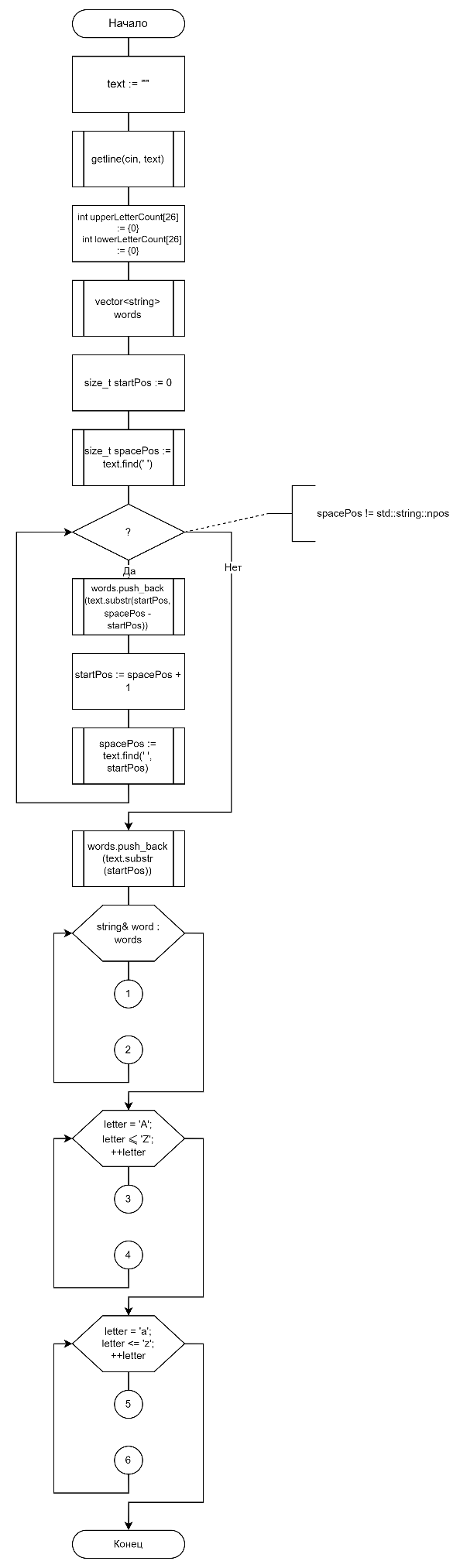


Рисунок 2 – Алгоритм решения задачи 2

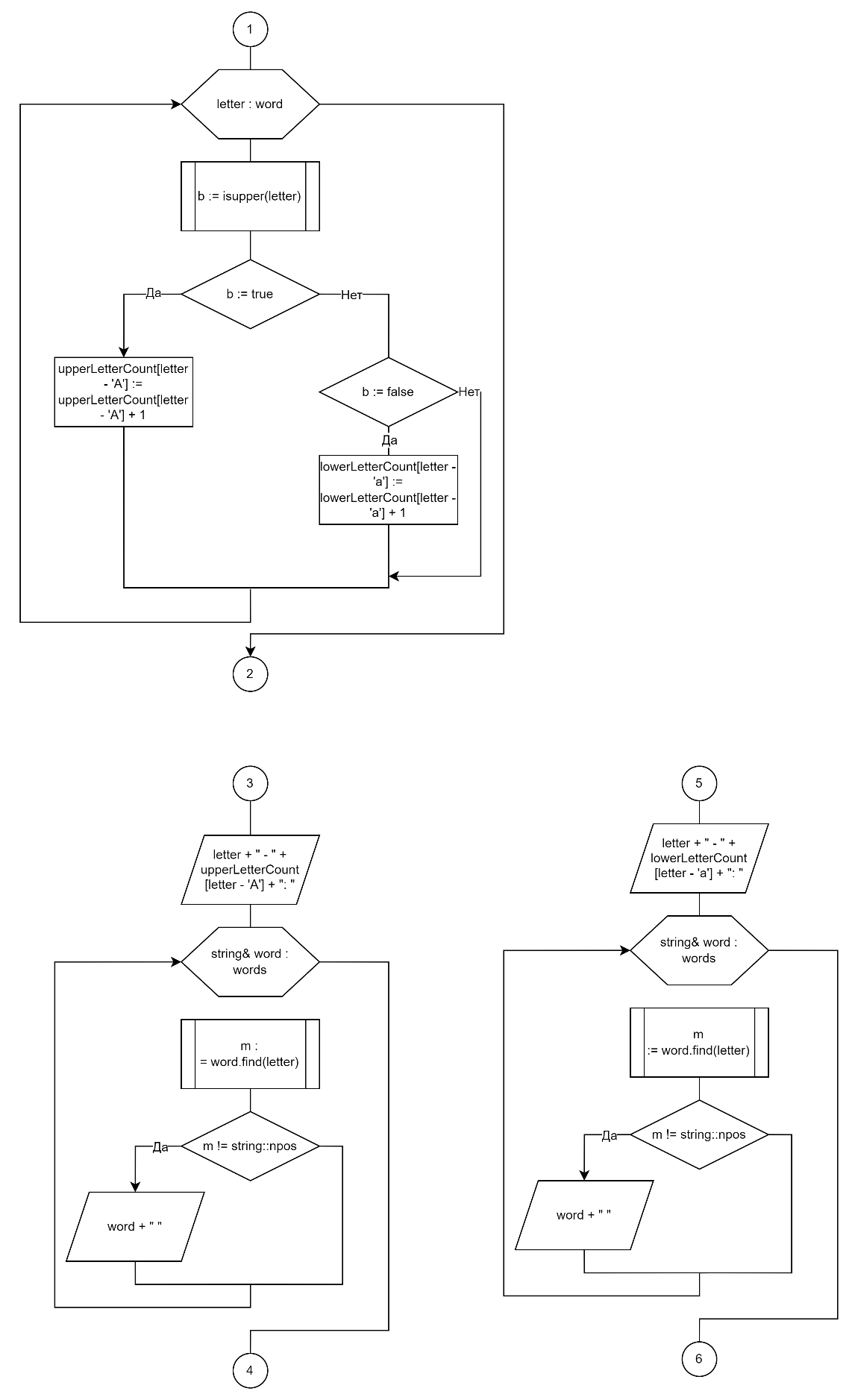


Рисунок 2 - Продолжение

*Задача 3*

Входные данные алгоритма решения задачи:

* inputText – строка.

Выходные данные алгоритма решения задачи:

* *words* – массив.

Алгоритм решения задачи представлен на рисунке 3.

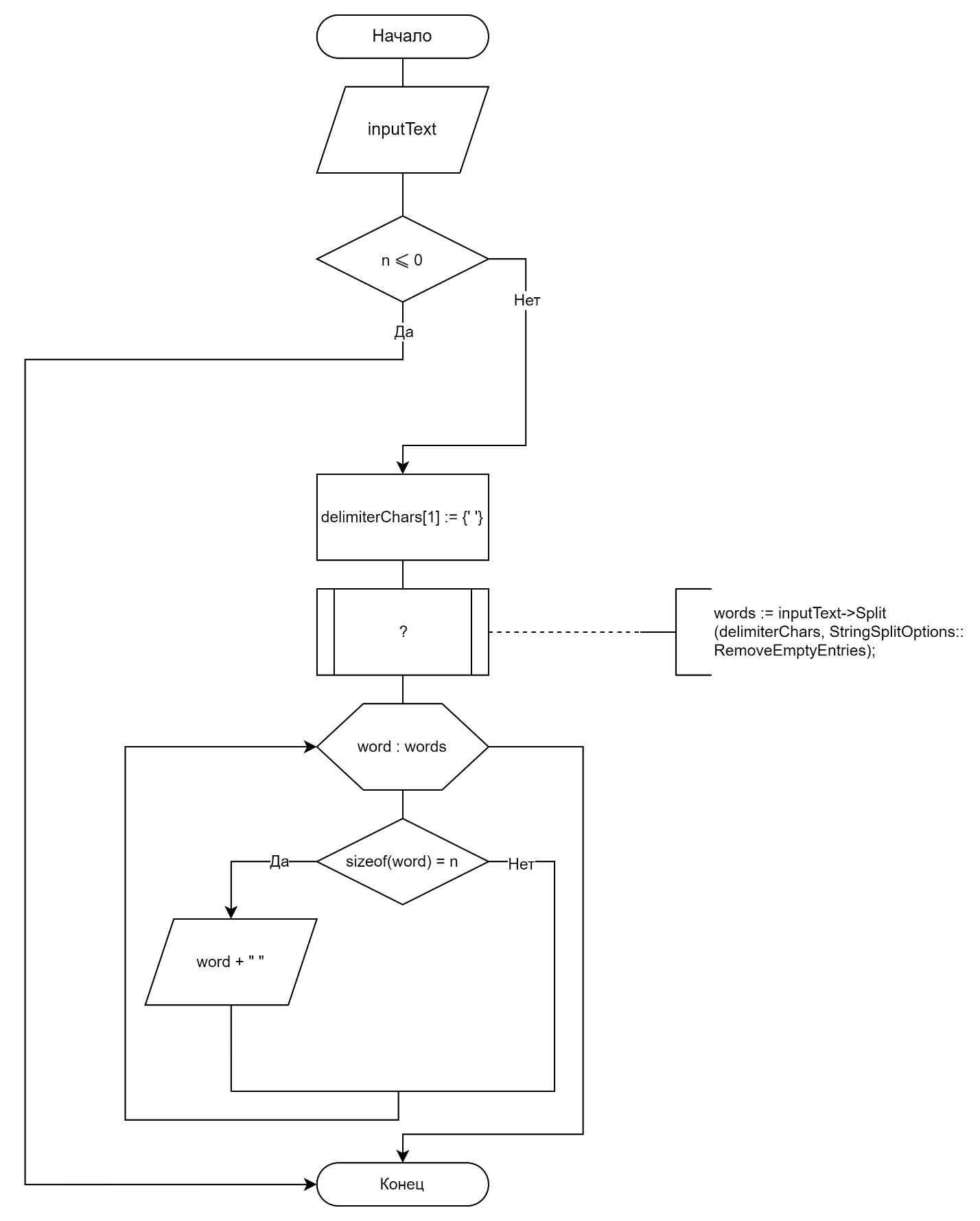


Рисунок 3 – Алгоритм решения задачи

***Текст программы***

*Текст программы для решения задачи 1*

// TODO: Дан текст. Определить частоту появления букв «а» в нем. Частота вычисляется как отношение количества данных символов в тексте к длине всего текста (пробелы учитываются, а символ конца строки не учитывается).

#include <iostream>

#include <string>

using namespace std;

int main(){

std::string inputText;

// Ввод текста

std::cout << "Input text: ";

std::getline(std::cin, inputText);

size\_t pos = inputText.find\_first\_of("aA");

size\_t countA3 = 0;

while (pos != std::string::npos) {

countA3++;

pos = inputText.find\_first\_of("aA", pos + 1);

}

// Вычисление длины текста (учитывая пробелы)

size\_t textLength = inputText.size();

// Проверка деления на ноль перед вычислением частоты

double frequency = (textLength > 0) ? static\_cast<double>(countA3) / textLength : 0.0;

// Вывод результата

std::cout << "Frequency of 'a': " << frequency << std::endl;

return 0;

}

*Текст программы для решения задачи 2*

// TODO: Дан текст на английском языке, слова в котором разделены одинарным символом пробела. Для каждой буквы алфавита вывести, в скольких словах этого текста она встречается с учетом регистра и сами слова (например, «a – 1: programming», «b – 0» и т. д).

#include <iostream>

#include <string>

#include <vector>

int main() {

std::cout << "Enter the text in english languge: " << std::endl;

std::string text = "";

getline(std::cin, text);

int upperLetterCount[26] = {0}; // Для букв верхнего регистра ('A' - 'Z')

int lowerLetterCount[26] = {0}; // Для букв нижнего регистра ('a' - 'z')

std::vector<std::string> words;

size\_t startPos = 0;

size\_t spacePos = text.find(' ');

while (spacePos != std::string::npos) {

words.push\_back(text.substr(startPos, spacePos - startPos));

startPos = spacePos + 1;

spacePos = text.find(' ', startPos);

}

words.push\_back(text.substr(startPos));

for (const std::string& word : words) {

for (char letter : word) {

if (std::isupper(letter)) {

upperLetterCount[letter - 'A']++;

} else if (std::islower(letter)) {

lowerLetterCount[letter - 'a']++;

}

}

}

for (char letter = 'A'; letter <= 'Z'; ++letter) {

std::cout << letter << " - " << upperLetterCount[letter - 'A'] << ": ";

for (const std::string& word : words) {

if (word.find(letter) != std::string::npos) {

std::cout << word << " ";

}

}

std::cout << std::endl;

}

for (char letter = 'a'; letter <= 'z'; ++letter) {

std::cout << letter << " - " << lowerLetterCount[letter - 'a'] << ": ";

for (const std::string& word : words) {

if (word.find(letter) != std::string::npos) {

std::cout << word << " ";

}

}

std::cout << std::endl;

}

return 0;

}}

*Текст программы для решения задачи 3*

// TODO: Дан текст, слова в котором разделены одинарным символом пробела, и целое n. Вывести на экран все слова длины n.

#include <iostream>

#using <mscorlib.dll>

using namespace System;

int main() {

// Считываем текст

Console::Write("Введите текст: ");

String^ inputText = Console::ReadLine();

// Считываем длину слова n

Console::Write("Введите длину слова n: ");

int n;

if (!Int32::TryParse(Console::ReadLine(), n) || n <= 0) {

Console::WriteLine("Ошибка: Введите корректное положительное число.");

return 1;

}

// Разбиваем текст на слова

array<Char>^ delimiterChars = {' '};

array<String^>^ words = inputText->Split(delimiterChars, StringSplitOptions::RemoveEmptyEntries);

// Выводим слова длины n

Console::Write("Слова длины {0}: ", n);

for each (String^ word in words) {

if (word->Length == n) {

Console::Write(word + " ");

}

}

return 0;

}

***Тестирование программы***

Тестирование задачи 1 представлено на рисунках 4, 5.



Рисунок 4 –Тест 1 задачи 1



Рисунок 5 – Тест 2 задачи 1

Тестирование задачи 2 представлено на рисунке 6.

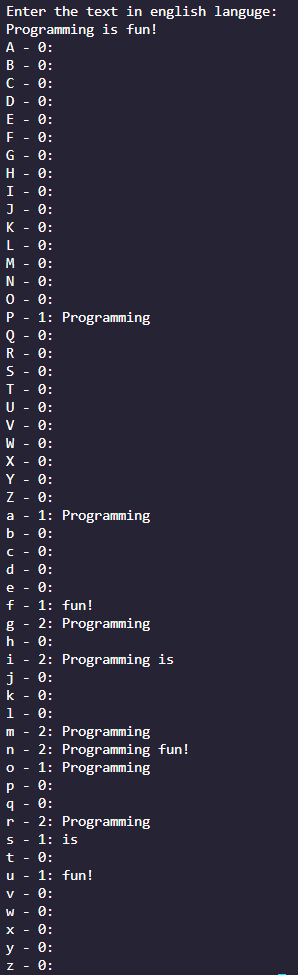


Рисунок 6 –Тест 1 задачи 2

Тестирование задачи 3 представлено на рисунке 7.



Рисунок 7 – Тест 1 Задачи 3