**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Звіт до лабораторної роботи № 26**

**з дисципліни**   
**«Програмування»**

**Виконав: Cтудент групи ІТ-23**

**Літвінов Кіріл**

**Варіант: 11**

**Перевірив:**

**1. Постановка завдання**

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

**2. Таблиця позначень**

|  |  |  |
| --- | --- | --- |
| **Позначення в програмі** | **Пояснення** | |
| bool isVowel(char c) | | |
| c = tolower(c); | використовується для забезпечення того, що порівняння символів буде вірним, незалежно від регістру | |
| bool startsAndEndsWithConsonant(const string& word) | | |
| char firstChar = tolower(word[0]); | Цей виклик використовується для перевірки, чи перший символ слова word є приголосною літерою. | |
| char lastChar = tolower(word[word.length() - 1]); | Цей виклик використовується для перевірки, чи останній символ слова word є приголосною літерою. | |
| string findLongestWord(const string& text) | | |
| string longestWord; | Це оголошення створює порожній об'єкт типу string, який поки що не містить жодних символів. | |
| string word; | Це оголошення створює порожній об'єкт типу string, який на початку не містить жодних символів. | |
| word += c; | Цей оператор використовується для поступового формування поточного слова word під час обробки тексту. | |
| longestWord = word; | У контексті даної програми, коли знайдено слово, яке є найдовшим до цього моменту, воно зберігається у змінній word. | |
| word = ""; | ця операція виконується для очищення значення змінної word. Після того, як поточне слово було оброблене | |
| longestWord = word; | коли знайдено слово, яке є найдовшим до цього моменту, воно зберігається у змінній word. | |
| void findWordsStartingAndEndingWithConsonant(const string& text, vector<string>& result) | | |
| result.push\_back(word); | коли знайдено слово, яке починається і закінчується на приголосну літеру, воно додається до вектора result. | |
| int main() | | |
| string text((istreambuf\_iterator<char>(inputFile)), istreambuf\_iterator<char>()); | цей рядок коду зчитує весь текст з файлу і зберігає його у рядковій змінній text. | |
| string longestWord = findLongestWord(text); | цей рядок коду знаходить найдовше слово в рядку text і зберігає його у змінну longestWord | |
| vector<string> wordsStartingAndEndingWithConsonant; | цей рядок коду створює порожній вектор wordsStartingAndEndingWithConsonant, який буде використовуватися для збереження слів, що відповідають заданій умові. | |
| findWordsStartingAndEndingWithConsonant(text, wordsStartingAndEndingWithConsonant); | це виклик функції findWordsStartingAndEndingWithConsonant з аргументами text і wordsStartingAndEndingWithConsonant. | |
| double percentage = static\_cast<double>(wordsStartingAndEndingWithConsonant.size()) / text.length() \* 100; | | обчислення відсоткового значення, яке представляє співвідношення кількості слів, які починаються і закінчуються на приголосну літеру |

**3. Текст програми**

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <cctype>

using namespace std;

// Функція, яка перевіряє, чи є символ голосним

bool isVowel(char c) {

c = tolower(c);

return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

}

// Функція, яка перевіряє, чи слово починається і закінчується на приголосну літеру

bool startsAndEndsWithConsonant(const string& word) {

if (word.empty()) {

return false;

}

char firstChar = tolower(word[0]);

char lastChar = tolower(word[word.length() - 1]);

return (!isVowel(firstChar) && !isVowel(lastChar) && isalpha(lastChar));

}

// Функція, яка знаходить найдовше слово в тексті

string findLongestWord(const string& text) {

string longestWord;

string word;

for (char c : text) {

if (isalpha(c) || c == '\'') {

word += c;

}

else {

if (word.length() > longestWord.length()) {

longestWord = word;

}

word = "";

}

}

if (word.length() > longestWord.length()) {

longestWord = word;

}

return longestWord;

}

// Функція, яка знаходить слова, які починаються і закінчуються на приголосну літеру

void findWordsStartingAndEndingWithConsonant(const string& text, vector<string>& result) {

string word;

for (char c : text) {

if (isalpha(c) || c == '\'') {

word += c;

}

else {

if (startsAndEndsWithConsonant(word)) {

result.push\_back(word);

}

word = "";

}

}

if (startsAndEndsWithConsonant(word)) {

result.push\_back(word);

}

}

int main() {

ifstream inputFile("text.txt");

if (!inputFile) {

cout << "Failed to open the input file." << endl;

return 1;

}

// Зчитування тексту з файлу

string text((istreambuf\_iterator<char>(inputFile)), istreambuf\_iterator<char>());

// Знаходження найдовшого слова

string longestWord = findLongestWord(text);

cout << "Longest word: " << longestWord << endl;

// Знаходження слів, які починаються і закінчуються на приголосну літеру

vector<string> wordsStartingAndEndingWithConsonant;

findWordsStartingAndEndingWithConsonant(text, wordsStartingAndEndingWithConsonant);

// Запис слів у файл "output.txt"

ofstream outputFile("output.txt");

if (!outputFile) {

cout << "Failed to open the output file." << endl;

return 1;

}

for (const string& word : wordsStartingAndEndingWithConsonant) {

outputFile << word << endl;

}

// Обчислення відсотка слів, які починаються і закінчуються на приголосну літеру, порівняно до всього тексту

double percentage = static\_cast<double>(wordsStartingAndEndingWithConsonant.size()) / text.length() \* 100;

cout << "Percentage: " << percentage << "%" << endl;

return 0;

}

**4. Приклад роботи**

Зображення, що містить текст, знімок екрана, Шрифт, чорний

Автоматично згенерований опис