Выполнил студент 1 курса ФИТУ группы 09.03.04 РПИб-о22 Ставничий Даниил Сергеевич.

Лабораторная работа №5

Задание: Дан большой текст в виде текстового файла, например, роман Толстого Л.Н. «Война и мир». Пользователь вводит запрос не короче 3 символов. Программа должна найти первые 20 слов из текста, содержащие в себе введённое слово. Первыми должны выводиться наиболее часто встречаемые в тексте слова (частота вхождения как индекс релевантности). Время на начальную обработку текста не более 1 минуты. Время обработки очередного запроса не более 2 с.

Код программы:

Основной файл - main.cpp

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <vector>

#include <unordered\_map>

#include <algorithm>

#include <chrono>

using *namespace* std;

// Определите структуру для хранения информации о частоте слов

*struct* WordFreq {

    string word;

*int* freq;

};

// Определите функцию для преобразования строки в слова

vector<string> tokenize(string *str*) {

    vector<string> tokens;

    stringstream ss(str);

    string token;

    while (ss >> token) {

        tokens.push\_back(token);

    }

    return tokens;

}

// Определите функцию для подсчета частоты слов в векторе

unordered\_map<string, *int*> countWords(vector<string> *words*) {

    unordered\_map<string, *int*> freqMap;

    for (string word : words) {

        freqMap[word]++;

    }

    return freqMap;

}

// Определите функцию для сравнения объектов WordFreq по частоте

*bool* compareWordFreq(const WordFreq& *a*, const WordFreq& *b*) {

    return a.freq > b.freq;

}

// Определите функцию для поиска слов в текстовом файле

vector<WordFreq> searchWords(string *filename*, string *query*) {

    // Открытие файла

    ifstream file(filename);

    if (!file.is\_open()) {

        cerr << "Error: could not open file " << filename << endl;

        return {};

    }

    // Считывание файла в строчку

    stringstream buffer;

    buffer << file.rdbuf();

    string text = buffer.str();

    // Преобразование текста в слова

    vector<string> words = tokenize(text);

    // Подсчёт частоты встречаемости слов

    unordered\_map<string, *int*> freqMap = countWords(words);

    // Сортировка слов по частоте

    vector<WordFreq> wordFreqs;

    for (*auto* it : freqMap) {

        wordFreqs.push\_back({it.first, it.second});

    }

    sort(wordFreqs.begin(), wordFreqs.end(), compareWordFreq);

    // Поиск по словам

    vector<WordFreq> results;

    for (WordFreq wordFreq : wordFreqs) {

        if (wordFreq.word.find(query) != string::npos) {

            results.push\_back(wordFreq);

            if (results.size() >= 20) {

                break;

            }

        }

    }

    // Закрыть файл

    file.close();

    return results;

}

*int* main() {

    // Получаем имя файла и запрос от пользователя

    string filename, query;

    cout << "Enter the filename: ";

    getline(cin, filename);

    cout << "Enter the query: ";

    getline(cin, query);

    // Поиск нужных слов и измерение времи

*auto* start = chrono::steady\_clock::now();

    vector<WordFreq> results = searchWords(filename, query);

*auto* end = chrono::steady\_clock::now();

*auto* diff = end - start;

*double* time\_ms = chrono::duration<*double*, milli>(diff).count();

    // Вывести результат

    cout << "Results:" << endl;

    for (WordFreq wordFreq : results) {

        cout << wordFreq.word << " (" << wordFreq.freq << ")" << endl;

    }

    cout << "Time: " << time\_ms << " ms" << endl;

    cout << "Работу выполнил: Ставничий Даниил Сергеевич, ФИТУ РПИб" << endl;

    return 0;

}

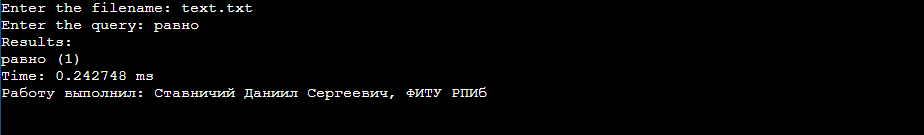
Файл с текстом – text.txt

Это утверждение указывает на то, что тип с плавающей точкой является неточным,и хотя double является наиболее точным типом среди всех типов с плавающей точкой в C++,он все равно не может обеспечить абсолютную точность.

Почему именно double? В C++ существует три типа с плавающей точкой: float, double и long double.Float обеспечивает меньшую точность, чем double, и может привести к потере значимых цифр после запятой.Long double обеспечивает большую точность, чем double, но в большинстве случаев он не является необходимым,а используется только в специальных случаях, где требуется наибольшая возможная точность.

В интонации возможно есть некоторая неуверенность,потому что операции с плавающей точкой являются сложными,и результаты могут зависеть от множества факторов,таких как алгоритмы округления, размерность чисел и т.д.Но при этом, тип double считается наиболее надежным и общепринятым типом с плавающей точкой в C++.

Результаты выполнения программы:



Вывод:

Данная программа соответствует описанию основного задания