

Міністерство освіти і науки, молоді та спорту України
Національний університет “Львівська політехніка”

Кафедра ЕОМ



Звіт

з домашнє завдання №19 з дисципліни:

“Алгоритми та моделі обчислень”

Варіант: № 24.

Виконав:

ст. групи КІ-203

Ширий Богдан Ігорович

Перевірив:

ст. викладач кафедри ЕОМ

Козак Назар Богданович

Львів – 2023

ЗАВДАННЯ:

УМОВА:

Застосовуючи STL скласти програму (C/C++), яка за допомогою *std::sort* дозволяє з вхідного тексту (*std::string*) вивести слова, що починається з заданої літери (решта слів вивести в алфавітному порядку). Сортування потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки *std::vector*, що зберігає індекси на початок слів. Додатково потрібно зберегти індекси на початок слів. Додатково потрібно зберегти результати роботи програми у *std::map*.

Індекси із *std::vector* вивести на екран за допомогою *std::copy*, а слова за допомогою звичайного циклу *for*. Для виводу даних з *std::map* застосувати *std::transform*.

ВИБІР ВАРІАНТУ:

Перша літера мого прізвища латинськими літерами це S.

ВИКОНАННЯ:

Склав C++ десктопну програму та її роботу зобразив на рисунках 1 та 2.

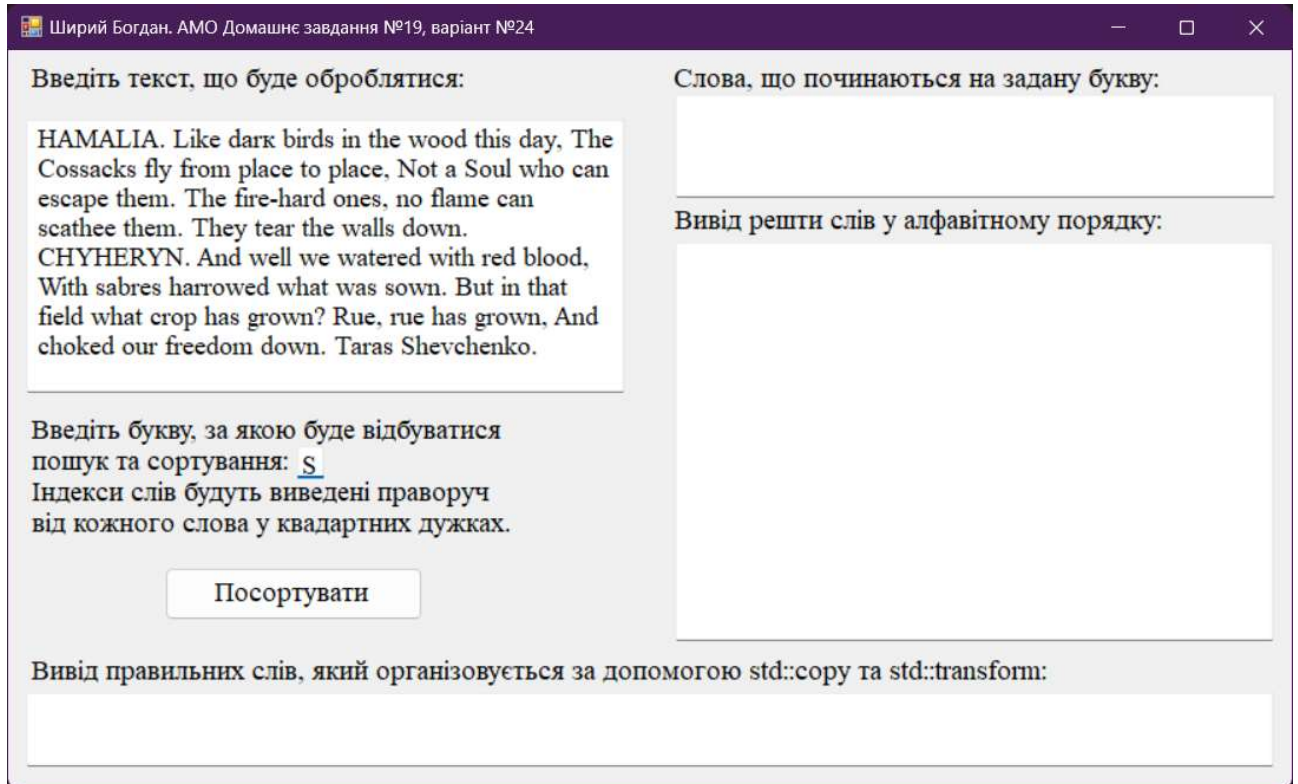


Рис. 1. Початкові дані програми.

Ця програма зчитує з текстбокса текст, для того щоб цілковито відповідати умові, не використовував спеціальних методів переводу типу даних таких як

System::Convert::ToString, а переводив з *System::String*[^] до *std::string* поелементно через цикл. Проте для виводу не зміг навіть спеціальних конверторів знайти, тому вирішив виведення організувати через файл. Файл використовується лише для виводу і тільки для нього – це можна побачити глянувши у код. Буквально, працював як з консольною програмою, проте замість *std::cout* використовував запис у файл, типу «*myfile << ...*», а потім записав дані з файлу у текстбокс виводу.

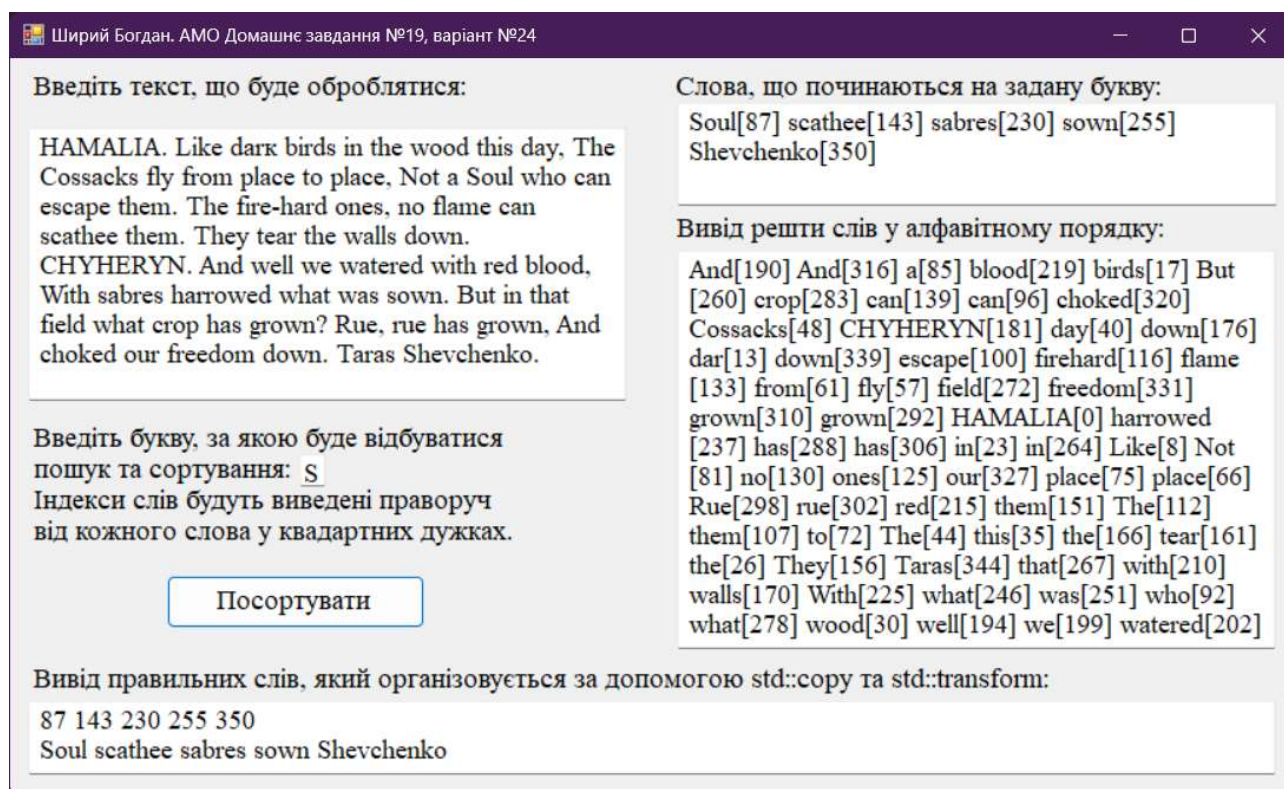


Рис. 2. Програма після натиску на кнопку.

Ця програма для роботи використовує лише *std::sort*, один *std::string*, один *std::vector* та один *std::map*. Як бачимо вона перше виводить усі слова (і ті, що починаються з великої літери, і ті, що починаються з маленької літери), які починаються на відповідну задану літеру, а решта слів вона виводить за алфавітним порядком, таким чином, щоб і малі, і великі літери виводилися разом, а не окремо. Також, нижче відповідно до умови організував вивід індексів із *std::vector* на екран за допомогою *std::copy*, де для виводу даних з *std::map* застосував *std::transform*. Ну і слова відповідно вивів за допомогою *for*.

Відповідно, у лістингах навів код кожного файлу:

- MainForm.cpp у лістингу 1,
- MainForm.h у лістингу 2.

Лістинг 1. Код файлу MainForm.cpp.

```
#include "MainForm.h"

using namespace System;
using namespace System::Windows::Forms;

[STAThreadAttribute]

int main() {
    Application::SetCompatibleTextRenderingDefault(false);
    Application::EnableVisualStyles();
    acmShyryiHW19::MainForm form;
    Application::Run(% form);
}
```

Лістинг 2. Код заголовкового файлу MainForm.h.

```
#pragma once
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <map>
#include <regex>
#include <fstream>

void sort_word_indices(std::vector<size_t>& words, const std::string& input, char starting_letter) {
    std::ofstream myfile;
    std::sort(words.begin(), words.end(), [&input](size_t a, size_t b) {
        return std::tolower(input[a]) < std::tolower(input[b]); // порівнюємо у порядку спадання
    });

    myfile.open("NotSelectedWords.txt"); // записую результат у файл
    for (size_t i = 0; i < words.size(); i++) {
        std::string word = input.substr(words[i], input.find(' ', words[i]) - words[i]);
        if (std::tolower(word[0]) != std::tolower(starting_letter)) {
            myfile << word << "[" << words[i] << "]" << " ";
        }
    }
    myfile.close();
}

struct GetSecond {
    const std::string& operator()(const std::pair<size_t, std::string>& p) const {
        return p.second;
    }
};

namespace acmShyryiHW19 {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::IO;

    /// <summary>
    /// Summary for MainForm
    /// </summary>
    public ref class MainForm : public System::Windows::Forms::Form
    {
    public:
        MainForm(void)
        {
            InitializeComponent();
            //
            //TODO: Add the constructor code here
            //
        }

    protected:
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        ~MainForm()
    {
    }
    };
}
```

```

        {
            if (components)
            {
                delete components;
            }
        }

private: System::Windows::Forms::TextBox^ inputTextBox;
private: System::Windows::Forms::Label^ inputInfoLabel;
private: System::Windows::Forms::Label^ inputLetterInfoLabel;
private: System::Windows::Forms::TextBox^ inputLetterTextBox;
private: System::Windows::Forms::TextBox^ outputSelectedWordsTextBox;
private: System::Windows::Forms::TextBox^ outputNotSelectedWordsTextBox;
private: System::Windows::Forms::Label^ outputSelectedWordsInfoLabel;
private: System::Windows::Forms::Label^ outputNotSelectedWordsInfoLabel;
private: System::Windows::Forms::TextBox^ outputOtherTextBox;
private: System::Windows::Forms::Button^ sortTextButton;
private: System::Windows::Forms::Label^ outputOtherInfoInfoLabel;
protected:

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->inputTextBox = (gcnew System::Windows::Forms::TextBox());
        this->inputInfoLabel = (gcnew System::Windows::Forms::Label());
        this->inputLetterInfoLabel = (gcnew System::Windows::Forms::Label());
        this->inputLetterTextBox = (gcnew System::Windows::Forms::TextBox());
        this->outputSelectedWordsTextBox = (gcnew System::Windows::Forms::TextBox());
        this->outputNotSelectedWordsTextBox = (gcnew System::Windows::Forms::TextBox());
        this->outputSelectedWordsInfoLabel = (gcnew System::Windows::Forms::Label());
        this->outputNotSelectedWordsInfoLabel = (gcnew System::Windows::Forms::Label());
        this->sortTextButton = (gcnew System::Windows::Forms::Button());
        this->outputOtherTextBox = (gcnew System::Windows::Forms::TextBox());
        this->outputOtherInfoInfoLabel = (gcnew System::Windows::Forms::Label());
        this->SuspendLayout();
        //
        // inputTextBox
        //
        this->inputTextBox->Location = System::Drawing::Point(12, 47);
        this->inputTextBox->Multiline = true;
        this->inputTextBox->Name = L"inputTextBox";
        this->inputTextBox->Size = System::Drawing::Size(406, 185);
        this->inputTextBox->TabIndex = 0;
        //
        // inputInfoLabel
        //
        this->inputInfoLabel->AutoSize = true;
        this->inputInfoLabel->Location = System::Drawing::Point(12, 9);
        this->inputInfoLabel->Name = L"inputInfoLabel";
        this->inputInfoLabel->Size = System::Drawing::Size(366, 26);
        this->inputInfoLabel->TabIndex = 1;
        this->inputInfoLabel->Text = L"Введіть текст, що буде оброблятися.";
        //
        // inputLetterInfoLabel
        //
        this->inputLetterInfoLabel->AutoSize = true;
        this->inputLetterInfoLabel->Location = System::Drawing::Point(12, 247);
        this->inputLetterInfoLabel->Name = L"inputLetterInfoLabel";
        this->inputLetterInfoLabel->Size = System::Drawing::Size(405, 104);
        this->inputLetterInfoLabel->TabIndex = 2;
        this->inputLetterInfoLabel->Text = L"Введіть букву, за якою буде відбуватися\ппошук та
        сортування:\пІндекси слів будуть виведені праворуч\пвід кожного слова у квадратних дужках.";
        //
        // inputLetterTextBox
        //
        this->inputLetterTextBox->Location = System::Drawing::Point(196, 268);
        this->inputLetterTextBox->Multiline = true;
        this->inputLetterTextBox->Name = L"inputLetterTextBox";
        this->inputLetterTextBox->Size = System::Drawing::Size(18, 22);
        this->inputLetterTextBox->TabIndex = 3;
        //
    }

```



```

// outputSelectedWordsTextBox
//
this->outputSelectedWordsTextBox->Location = System::Drawing::Point(452, 30);
this->outputSelectedWordsTextBox->Multiline = true;
this->outputSelectedWordsTextBox->Name = L"outputSelectedWordsTextBox";
this->outputSelectedWordsTextBox->Size = System::Drawing::Size(407, 70);
this->outputSelectedWordsTextBox->TabIndex = 4;
//
// outputNotSelectedWordsTextBox
//
this->outputNotSelectedWordsTextBox->Location = System::Drawing::Point(452, 130);
this->outputNotSelectedWordsTextBox->Multiline = true;
this->outputNotSelectedWordsTextBox->Name = L"outputNotSelectedWordsTextBox";
this->outputNotSelectedWordsTextBox->Size = System::Drawing::Size(406, 270);
this->outputNotSelectedWordsTextBox->TabIndex = 5;
//
// outputSelectedWordsInfoLabel
//
this->outputSelectedWordsInfoLabel->AutoSize = true;
this->outputSelectedWordsInfoLabel->Location = System::Drawing::Point(447, 9);
this->outputSelectedWordsInfoLabel->Name = L"outputSelectedWordsInfoLabel";
this->outputSelectedWordsInfoLabel->Size = System::Drawing::Size(410, 26);
this->outputSelectedWordsInfoLabel->TabIndex = 6;
this->outputSelectedWordsInfoLabel->Text = L"Слова, що починаються на задану букву:";
//
// outputNotSelectedWordsInfoLabel
//
this->outputNotSelectedWordsInfoLabel->AutoSize = true;
this->outputNotSelectedWordsInfoLabel->Location = System::Drawing::Point(448, 105);
this->outputNotSelectedWordsInfoLabel->Name = L"outputNotSelectedWordsInfoLabel";
this->outputNotSelectedWordsInfoLabel->Size = System::Drawing::Size(418, 26);
this->outputNotSelectedWordsInfoLabel->TabIndex = 7;
this->outputNotSelectedWordsInfoLabel->Text = L"Вивід решти слів у алфавітному
порядку:";
//
// sortTextButton
//
this->sortTextButton->Location = System::Drawing::Point(106, 350);
this->sortTextButton->Name = L"sortTextButton";
this->sortTextButton->Size = System::Drawing::Size(175, 36);
this->sortTextButton->TabIndex = 8;
this->sortTextButton->Text = L"Посортувати";
this->sortTextButton->UseVisualStyleBackColor = true;
this->sortTextButton->Click += gcnew System::EventHandler(this,
&MainForm::sortTextButton_Click);
//
// outputOtherTextBox
//
this->outputOtherTextBox->Location = System::Drawing::Point(12, 435);
this->outputOtherTextBox->Multiline = true;
this->outputOtherTextBox->Name = L"outputOtherTextBox";
this->outputOtherTextBox->Size = System::Drawing::Size(846, 50);
this->outputOtherTextBox->TabIndex = 4;
//
// outputOtherInfoInfoLabel
//
this->outputOtherInfoInfoLabel->AutoSize = true;
this->outputOtherInfoInfoLabel->Location = System::Drawing::Point(12, 410);
this->outputOtherInfoInfoLabel->Name = L"outputOtherInfoInfoLabel";
this->outputOtherInfoInfoLabel->Size = System::Drawing::Size(410, 26);
this->outputOtherInfoInfoLabel->TabIndex = 6;
this->outputOtherInfoInfoLabel->Text = L"Вивід правильних слів, який організовується за
допомогою std::copy та std::transform:";
//
// MainForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(13, 26);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(870, 500);
this->Controls->Add(this->outputOtherInfoInfoLabel);
this->Controls->Add(this->outputOtherTextBox);
this->Controls->Add(this->sortTextButton);
this->Controls->Add(this->outputNotSelectedWordsInfoLabel);
this->Controls->Add(this->outputSelectedWordsInfoLabel);
this->Controls->Add(this->outputNotSelectedWordsTextBox);
this->Controls->Add(this->outputSelectedWordsTextBox);
this->Controls->Add(this->inputLetterTextBox);
this->Controls->Add(this->inputLetterInfoLabel);
this->Controls->Add(this->inputInfoLabel);
this->Controls->Add(this->inputTextBox);

```

```

        this->Font = (gcnew System::Drawing::Font(L"Times New Roman", 13.8F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
        this->Margin = System::Windows::Forms::Padding(5);
        this->Name = L"MainForm";
        this->Text = L"Ширий Богдан. АМО Домашнє завдання %19, варіант %24";
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
private: System::Void sortTextButton_Click(System::Object^ sender, System::EventArgs^ e) {
    std::ofstream myfile;
    // Переводимо типи даних з System в std
    std::string input(inputTextBox->Text->Length, ' ');
    for (int i = 0; i < inputTextBox->Text->Length; i++)
    {
        input[i] = static_cast<char>(inputTextBox->Text[i]);
    }
    input = std::regex_replace(input, std::regex("[^a-zA-Z0-9\\s]"), "");

    char starting_letter = inputLetterTextBox->Text[0];

    // Робота самої програми
    std::vector<size_t> word_indices;
    for (size_t i = 0; i < input.length(); i++) {
        if (i == 0 || input[i - 1] == ' ') {
            word_indices.push_back(i);
        }
    }

    myfile.open("SelectedWords.txt"); // записую результат у файл
    for (const auto& i : word_indices) {
        std::string word = input.substr(i, input.find(' ', i) - i);
        if (std::tolower(word[0]) == std::tolower(starting_letter)) {
            myfile << word << "[" << i << "]" << " ";
        }
    }
    myfile.close();

    // Використовую функцію, яка сортує зі std::sort, оскільки у цьому блоці не
    sort_word_indices(word_indices, input, starting_letter); // можна використовувати лямбда вирази

    // Робота самої програми
    word_indices.clear();
    for (size_t i = 0; i < input.length(); i++) {
        if ((i == 0 || input[i - 1] == ' ') && std::tolower(input[i]) ==
std::tolower(starting_letter)) {
            word_indices.push_back(i);
        }
    }

    // створення std::map
    std::map<size_t, std::string> word_map;
    for (const auto& i : word_indices) {
        std::string word = input.substr(i, input.find(' ', i) - i);
        if (std::tolower(word[0]) == std::tolower(starting_letter)) {
            word_map[i] = word;
        }
    }

    // Виведення слів та індексів на екран
    myfile.open("Another.txt"); // записую результат у файл
    std::copy(word_indices.begin(), word_indices.end(), std::ostream_iterator<size_t>(myfile, "
"));
    myfile << std::endl;
    std::transform(word_map.begin(), word_map.end(), std::ostream_iterator<std::string>(myfile, "
"), GetSecond());
    myfile.close();

    // Запис результатів з файлу у TextBox
    StreamReader^ reader = gcnew StreamReader("SelectedWords.txt");
    outputSelectedWordsTextBox->Text = reader->ReadToEnd();
    reader = gcnew StreamReader("NotSelectedWords.txt");
    outputNotSelectedWordsTextBox->Text = reader->ReadToEnd();
    reader = gcnew StreamReader("Another.txt");
    outputOtherTextBox->Text = reader->ReadToEnd();
}
};
}

```