

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

Кафедра ЕОМ



Звіт

з домашнє завдання №13 з дисципліни:

“Алгоритми та моделі обчислень”

Варіант: № 24.

Виконав:

ст. групи КІ-203

Ширий Богдан Ігорович

Перевірів:

ст. викладач кафедри ЕОМ

Козак Назар Богданович

Львів – 2023

ЗАВДАННЯ:

УМОВА:

Скласти програму (C/C++), яка за допомогою швидкого сортування (з стандартної бібліотеки мови C) дозволяє з вхідного тексту вивести слова, що починаються з заданої літери (решта слів вивести в алфавітному порядку). Сортування потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки масив, що зберігає вказівники на початок слів.

ВИБІР ВАРІАНТУ:

Перша літера мого прізвища латинськими літерами це S.

ВИКОНАННЯ:

Склав C++ десктопну програму та у лістингах навів код кожного файлу:

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

Відповідно, виконання програми та її вивід зобразив на рисунках від 1 до 2.

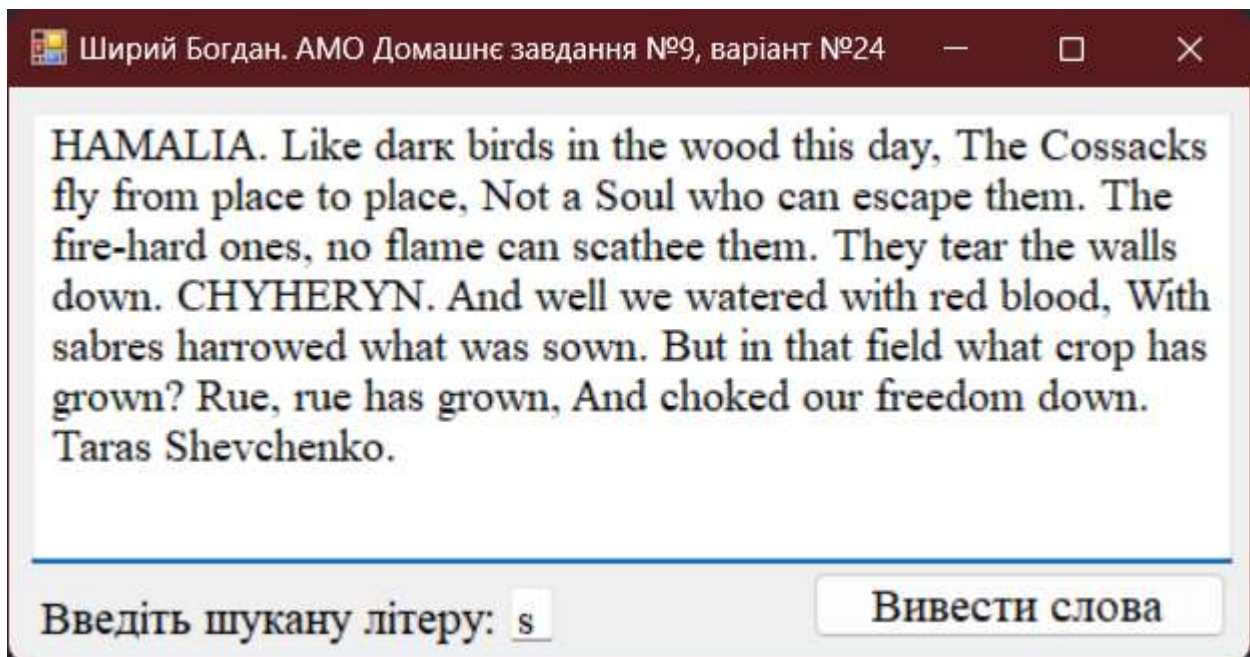


Рис. 1. Початкові дані програми.

Ця програма зчитує з текстбокса текст, для того щоб цілковито відповідати умові не використовував спеціальних методів перевodu типу даних таких як System::Convert::ToString, а переводив з System::String^ до char* поелементно через цикл. Таким самим чином організував перепис текстбокса вже з правильним порядком слів.

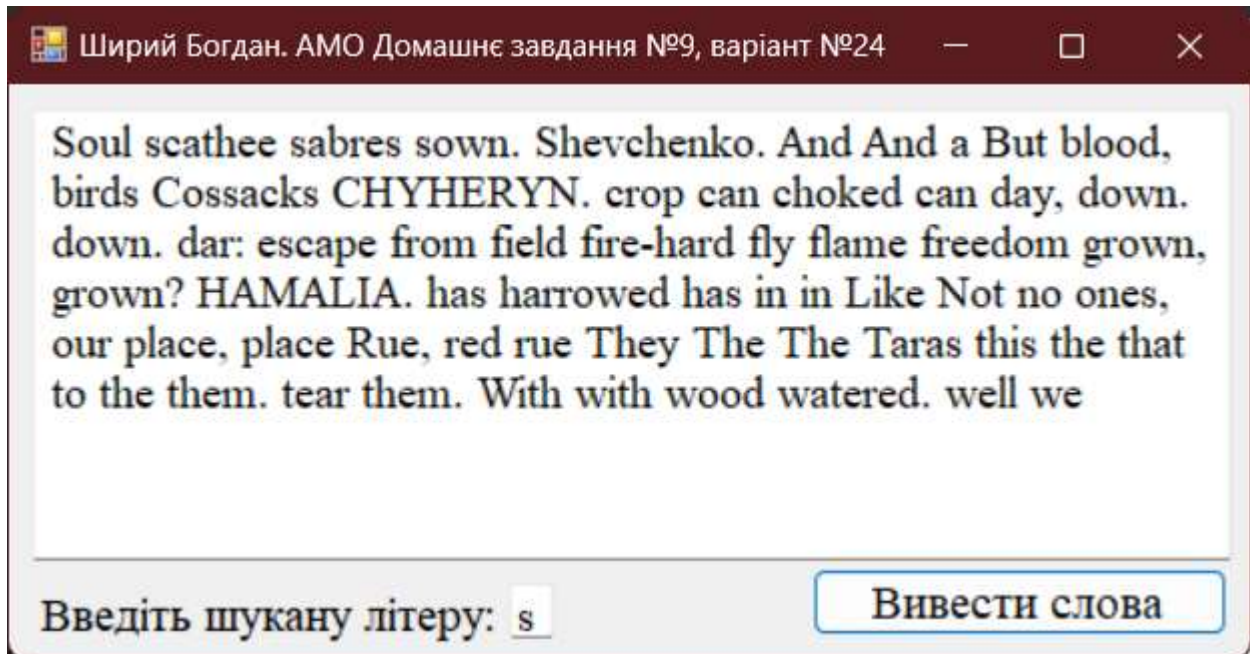


Рис. 2. Програма після натиску на кнопку.

Ця програма для роботи використовує лише швидке сортування та масив, що зберігає вказівники на початок слів для роботи. Як бачимо вона перше виводить усі слова (і ті, що починаються з великої літери, і ті, що починаються з великої 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.

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();
    acmShyryiHW13::MainForm form;
    Application::Run(% form);
}
```

Лістинг 2. Код заголовкового файлу MainForm.h.

```
#pragma once

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>

using namespace std;

bool startsWith(const char* str, char c) {
    return str[0] == c || str[0] == c - 32 || str[0] == c + 32;
}

bool compareWords(const char* a, const char* b) {
    int result = strcmp(a, b);
    if (result == 0) {
        return false;
    }
    char aLower = tolower(a[0]);
    char bLower = tolower(b[0]);
    if (aLower == bLower) {
        return a[0] < b[0];
    }
    return aLower < bLower;
}

void quicksort(const char** arr, int left, int right) {
    if (left >= right) {
        return;
    }

    int i = left;
    int j = right;
    const char* pivot = arr[(left + right) / 2];

    while (i <= j) {
        while (compareWords(arr[i], pivot)) {
            i++;
        }
        while (compareWords(pivot, arr[j])) {
            j--;
        }
        if (i <= j) {
            swap(arr[i], arr[j]);
            i++;
            j--;
        }
    }

    quicksort(arr, left, j);
    quicksort(arr, i, right);
}

namespace acmShyryiHW13 {

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

    /// <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::Button^ findLetterButton;
private: System::Windows::Forms::Label^ infoLabel;
private: System::Windows::Forms::TextBox^ letterTextBox;
protected:

private: System::ComponentModel::IContainer^ components;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>

#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->findLetterButton = (gcnew System::Windows::Forms::Button());
        this->infoLabel = (gcnew System::Windows::Forms::Label());
        this->letterTextBox = (gcnew System::Windows::Forms::TextBox());
        this->SuspendLayout();
        //
        // inputTextBox
        //
        this->inputTextBox->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->inputTextBox->Location = System::Drawing::Point(12, 12);
        this->inputTextBox->Multiline = true;
        this->inputTextBox->Name = L"inputTextBox";
        this->inputTextBox->Size = System::Drawing::Size(638, 220);
        this->inputTextBox->TabIndex = 0;
        //
        // findLetterButton
        //
        this->findLetterButton->Font = (gcnew System::Drawing::Font(L"Times New
Roman", 13.8F));
        this->findLetterButton->Location = System::Drawing::Point(428, 238);
        this->findLetterButton->Name = L"findLetterButton";
        this->findLetterButton->Size = System::Drawing::Size(222, 33);
        this->findLetterButton->TabIndex = 1;
        this->findLetterButton->Text = L"Вивести слова";
        this->findLetterButton->UseVisualStyleBackColor = true;
        this->findLetterButton->Click += gcnew System::EventHandler(this,
&MainForm::findLetterButton_Click);
        //
        // infoLabel
        //
        this->infoLabel->AutoSize = true;
        this->infoLabel->Font = (gcnew System::Drawing::Font(L"Times New Roman",
13.8F));
        this->infoLabel->Location = System::Drawing::Point(12, 248);
        this->infoLabel->Name = L"infoLabel";
        this->infoLabel->Size = System::Drawing::Size(236, 26);
        this->infoLabel->TabIndex = 2;
        this->infoLabel->Text = L"Введіть шукану літеру.";
        //
        // letterTextBox
        //
        this->letterTextBox->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->letterTextBox->Location = System::Drawing::Point(266, 245);
        this->letterTextBox->Multiline = true;
        this->letterTextBox->Name = L"letterTextBox";
        this->letterTextBox->Size = System::Drawing::Size(23, 26);
        this->letterTextBox->TabIndex = 3;
        //
    }

```

```

        // MainForm
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(662, 283);
        this->Controls->Add(this->letterTextBox);
        this->Controls->Add(this->infoLabel);
        this->Controls->Add(this->findLetterButton);
        this->Controls->Add(this->inputTextBox);
        this->Name = L"MainForm";
        this->Text = L"Ширій Богдан. АМО Домашнє завдання %9, варіант %24";
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
    private: System::Void findLetterButton_Click(System::Object^ sender, System::EventArgs^ e) {
        try
        {
            const int MAX_SIZE = 1000;
            char text[MAX_SIZE];
            char inputLetter = letterTextBox->Text[0];

            for (int i = 0; i < inputTextBox->Text->Length - 4; i++)
            {
                text[i] = inputTextBox->Text[i];
            }

            char letter = tolower(inputLetter); // перетворюємо введенний символ до
            нижнього регістру

            const char* words[MAX_SIZE];
            int wordsCount = 0;

            char* token = strtok(text, " ");
            while (token != nullptr) {
                words[wordsCount++] = token;
                token = strtok(nullptr, " ");
            }

            int partitionIndex = 0;
            for (int i = 0; i < wordsCount; i++) {
                if (startsWith(words[i], tolower(letter)) || startsWith(words[i],
                toupper(letter))) {
                    swap(words[i], words[partitionIndex]);
                    partitionIndex++;
                }
            }

            quicksort(words, partitionIndex, wordsCount - 1);

            inputTextBox->Text = "";
            for (int i = 0; i < wordsCount; i++) {
                inputTextBox->Text += gcnew System::String(words[i]) + " ";
            }
        }
        catch (System::Exception^ e) {}
    }
};
}

```