**зМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра «Інженерії програмного забезпечення»**

****

**Лабораторна робота №2**

Дисципліна "Структури та організація даних"

**Виконав:**

**Студент групи 201**

Грабовський.Є.О.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

**Викладач**

Кірей К.О.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

**Миколаїв – 2020**

**Задача 2.1. Пошук підрядка**

*Лістинг коду:*

void firstExV2() {

    std::ifstream is("text\_city\_ukr\_ansi.txt");

    if (!is.is\_open()) {

        std::cout << "Opens error\n";

        return;

    }

    is.seekg(0, std::ios::end);

    int length = is.tellg();

    is.seekg(0, std::ios::beg);

    char\* buffer = new char[length];

    char\* word = new char[length];

    std::cin.getline(word, length);

    int lenghtWord = std::cin.tellg();

    --lenghtWord;

    --length;

    word[lenghtWord] = 0;

    is.read(buffer, length);

    buffer[length] = '\0';

    is.close();

    for (int index1 = 0, index2, index3; buffer[index1] != 0; ++index1) {

        for (index2 = index1, index3 = 0; word[index3] != 0 || buffer[index2] != 0; ++index2, ++index3)

        {

            if (buffer[index2] != word[index3])

            {

                while (buffer[index2] != '\n' && buffer[index2] != 0) {

                    index2++;

                }

                index1 = index2;

                break;

            }

            if (word[index3 + 1] == 0 || word[index3 + 1] == '\n') {

                std::cout << "Finded\n";

                return;

            }

        }

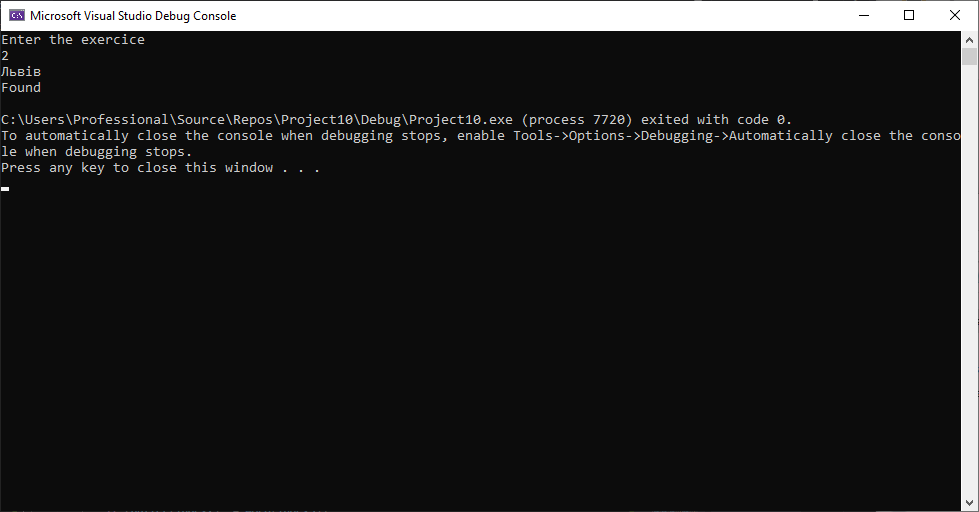
    }

    delete[] word;

    delete[] buffer;

}

Результат виконання програми:



**Задача 2.2. Обчислення кількості входжень слова у тексті**

*Лістинг коду:*

void secondEx() {

    std::ifstream is("Shakespeare\_Hamlet.txt");

    int length, wordLength;

    is.seekg(0, std::ios::end);

    length = is.tellg();

    is.seekg(std::ios::beg);

    char\* buffer = new char[length];

    is.read(buffer, length);

    char\* word = new char[length];

    std::cin.getline(word, length);

    buffer[length] = 0;

    wordLength = std::cin.tellg();

    wordLength -= 2;

    word[wordLength] = 0;

    //массив сепараторов, нужен для обозначения конца слова

    char separators[]{ '.', ',','!','?',';', '\n', ' ', '\0' };

    int separatorsLenght = 7;

    int count = 0;

    for (int index = 0; index < length;) {

        for (int j = 0; j < wordLength; ++j, ++index) {

            //если одна буква не совпала - идёт перебор елементов, пока не попадётся какой-либо из сепараторов

            if (word[j] != buffer[index]) {

                for (int i = 0; true; ++i) {

                    if (i == separatorsLenght)

                    {

                        i = 0;

                        index++;

                    }

                    if (buffer[index] == separators[i])

                    {

                        index++;

                        break;

                    }

                }

                break;

            }

            if (j + 1 == wordLength) {

                bool checker = false;

                index++;

                /\*проверка на то, является ли слово из буфера таким же

                    если следующий символ один из сепараторов, то да, это слово такое же

                    если нет, то значит оно больше чем то, которое задали

\*/

                for (int i = 0; ; ++i) {

                    if (buffer[index] == separators[i])

                    {

                        index++;

                        checker = true;

                        break;

                    }

                    if (i + 1 == separatorsLenght) {

                        index++;

                        break;

                    }

                }

                if (checker)

                    count++;

                break;

            }

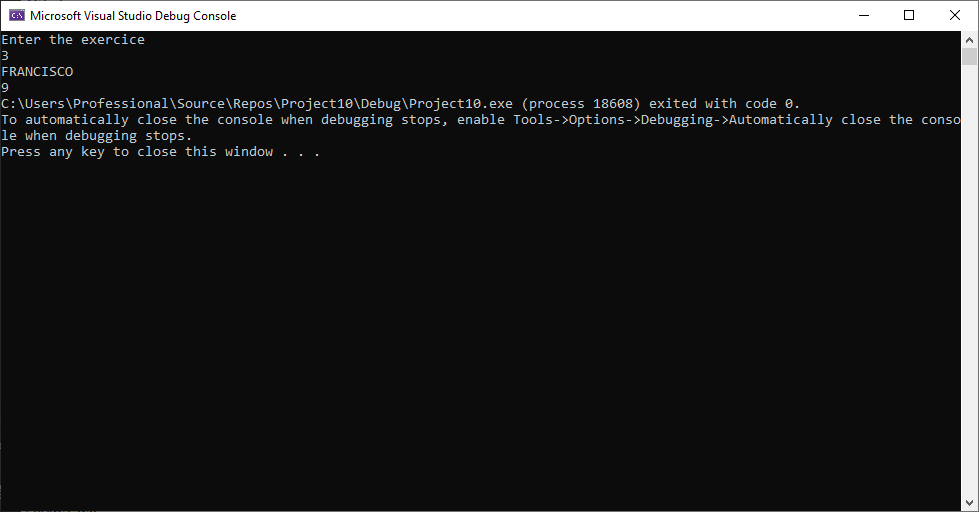
        }

    }

    std::cout << count;

}

Результат виконання програми:



**Вірішити попередню задачу за допомогою strtok**

void secondExV2() {

    std::ifstream is("Shakespeare\_Hamlet.txt");

    int length, wordLength;

    is.seekg(0, std::ios::end);

    length = is.tellg();

    is.seekg(std::ios::beg);

    char\* buffer = new char[length];

    is.read(buffer, length);

    char\* word = new char[length];

    std::cin.getline(word, length);

    buffer[length] = 0;

    wordLength = std::cin.tellg();

    wordLength -= 2;

    word[wordLength] = 0;

    int count = 0;

    char\* pch;

    char separators[]{ '.', ',','!','?',';', '\n', ' ', '\0' };

    pch = strtok(buffer, separators);

    bool checker;

    while (pch != NULL)

    {

        checker = true;

        for (int index = 0; pch[index] != '\0'; ++index) {

            if (pch[index] != word[index] || (pch[index + 1] == '\0' && word[index + 1] != '\0')) {

                checker = false;

                break;

            }

        }

        pch = strtok(NULL, separators);

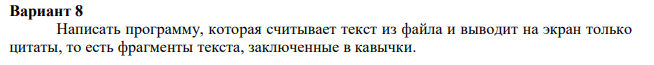
        if (checker) count++;

    }

    std::cout << count;

}

**Задача 2.3 Варіант 8**

*Лістинг коду:*

void thirdEx() {

    std::ifstream is("text.txt");

    int lenght;

    is.seekg(0, std::ios::end);

    lenght = is.tellg();

    is.seekg(0, std::ios::beg);

    char\* arr = new char[lenght];

    is.read(arr, lenght);

    --lenght;

    arr[lenght] = 0;

    is.close();

    bool fIn = false;

    int index = 0;

    while (arr[index] != '\0') {

        if (arr[index] == '\"' && fIn) {

            fIn = false;

            std::cout << arr[index];

            index++;

        }

        if (fIn) {

            std::cout << arr[index];

        }

        if (arr[index] == '\"' && !fIn) {

            fIn = true;

            std::cout << arr[index];

        }

        ++index;

    }

}

Результат виконання програми:

Вихідий рядок останнього завдання - "Hello?" Are "No!"

