

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №9

за курсом “Програмування”

студентки групи ПА-23-1

Мазур Віолети

кафедра комп’ютерних технологій, ДНУ

2023/2024

1. Постановка задачі

1 завдання

Обробка текстової інформації, що міститься у масиві символів. Виконати задані дії із введенним текстом :

11. Вивести всі слова, які повторюються та число їх повторень.

2 завдання

Рядкові функції.

Завдання: написати програму, в якій були б розроблені 2 функції, що імітують дії стандартних рядкових функцій (див. індивідуальне завдання). Вибір функцій здійснюється за допомогою пунктів меню.

Варіант №11 `strchr()` `strncmp()`

3 завдання

Обробка текстової інформації, що міститься у файлах.

Завдання:

1. За допомогою текстового редактора створити файл, що містить текст. Довжина рядка тексту не повинна перевищувати 80 символів. Це вхідний файл.

2. Програма повинна 1) інформацію вхідного файлу записати у вихідний файл; 2) реалізувати дії, вказані в індивідуальному завданні і занести відповідну інформацію у вихідний файл.
3. Ім'я вхідного файлу задає користувач в командному рядку або під час виконання програми.
4. Ім'я вихідного файлу: перші символи - не більше трьох символів з імені вхідного файлу, а решти символів _out.
5. Вихідний файл повинен мати розширення .dat.
6. Програма повинна забезпечувати по запиту користувача: 1) виведення тексту вхідного файлу на екран дисплея; 2) виведення тексту вихідного файлу на екран дисплея.
7. Закінчені послідовності дій оформити у вигляді функцій. Всі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.
8. Для вводу і виводу інформації (і для консолі, і для текстових файлів) застосувати засоби форматowanego вводу/виводу.
9. Визначення функцій і головну функцію розмістити в двох окремих файлах.

Варіант №11. Визначити кількість цифр в вхідному файлі, замінити цифри відповідними словами.

2. Опис розв'язку

1 завдання

Цей програмний код призначений для обчислення кількості повторень кожного введенного користувачем слова і їх виведення на екран. Ось опис кожної частини програми:

Функція ``areEqual``:

- Ця функція приймає два масиви символів ``arr1`` та ``arr2`` і їхні розміри ``n``.

- Вона порівнює елементи обох масивів і повертає `'true'`, якщо вони ідентичні, і `'false'` у протилежному випадку.

Головна функція `'main'`:

- У цій функції встановлюється кодова сторінка консолі для підтримки введення та виведення українських символів (кодова сторінка 1251).

- Визначається константа `'MAX_WORDS'`, яка представляє максимальну кількість слів, яку може ввести користувач.

- Створюється масив `'words'`, що містить до `'MAX_WORDS'` слів, кожне з яких може містити до 19 символів та додатковий символ завершення рядка `'\0'`.

- Користувачу пропонується ввести слова, доки він не введе символ "+" або не буде досягнуто максимальної кількості слів.

- Після введення слів виконується перевірка на кількість повторень кожного слова.

- Для цього використовується вбудована функція `'strcmp'` для порівняння двох рядків.

- Якщо слово вже пораховане, воно пропускається.

- В іншому випадку використовується вкладений цикл для підрахунку кількості повторень слова.

- Результати виводяться на екран у вигляді "Слово: <слово> Кількість: <кількість повторень>".

Цей код дозволяє користувачеві вводити слова та підраховує кількість їх повторень у введеному тексті.

2 завдання

Цей код реалізує дві функції, що імітують поведінку стандартних бібліотечних функцій `'strchr'` і `'strncmp'`. Крім того, він містить меню для користувача, що дозволяє викликати ці функції і порівнювати результати з відповідними стандартними функціями. Ось опис, що пояснює використані частини коду:

- `#include <iostream>`: Підключення бібліотеки для роботи з вхідними і вихідними потоками.
- `#include <cstring>`: Підключення бібліотеки для роботи з рядками С-стилю.
- `#include <windows.h>`: Підключення бібліотеки для використання функцій Windows API (зміна кодування консолі).
- `using namespace std`: Використання стандартного простору імен, що дозволяє уникнути явного зазначення `std::`.

Реалізація функцій

``my_strchr``

```
char* my_strchr(const char* str, char ch) {
    char* lastOccurrence = nullptr;
    while (*str) {
        if (*str == ch) {
            lastOccurrence = const_cast<char*>(str);
        }
        str++;
    }
    return lastOccurrence;
}
```

- Ця функція шукає останнє входження символу ``ch`` у рядку ``str``.
- Використовується вказівник ``lastOccurrence`` для збереження адреси останнього знайденого символу.
- Коли символ знайдений, вказівник оновлюється.
- Після проходження всього рядка повертається вказівник на останнє входження символу або ``nullptr``, якщо символ не знайдено.

``my_strncmp``

```

int my_strncmp(const char* str1, const char* str2, int n) {
    int cn = 0;
    while (*str1 && *str2 && cn < n) {
        if (*str1 != *str2) {
            return *str1 - *str2;
        }
        str1++;
        str2++;
        cn++;
    }

    if (cn == n) {
        return 0;
    }

    if (*str1) {
        return 1;
    } else if (*str2) {
        return -1;
    } else {
        return 0;
    }
}

```

- Ця функція порівнює до `n` символів двох рядків `str1` і `str2`.

- Повертає різницю між першими непарними символами, або 0, якщо рядки рівні до n символів.
- Якщо один з рядків коротший, повертається відповідний результат (-1 або 1).

Основна програма

- Основна програма змінює кодування консолі на 1251 для підтримки кирилиці.
- Меню дозволяє користувачеві вибрати одну з опцій: виклик `my_strchr`, виклик `my_strncmp`, або вихід з програми.
- У кожній опції програма запитує у користувача необхідні дані, викликає відповідну функцію, і порівнює результат з аналогічною стандартною функцією.
- Після виконання кожної функції програма виводить результати і порівняння на екран.
- Цикл `do-while` забезпечує повторне відображення меню до тих пір, поки користувач не вибере вихід.

3 завдання

Даний код є програмою для обробки текстових файлів, яка дозволяє користувачу виконувати різні операції над вхідним файлом. Нижче наведено детальний опис кожної частини коду, що пояснює призначення та використання кожного блоку коду.

1. Бібліотеки

- `<iostream>`: Використовується для вводу/виводу в консоль.
- `<fstream>`: Використовується для роботи з файлами.
- `<cstring>`: Використовується для роботи з C-рядками.
- `<cstdlib>`: Використовується для стандартних бібліотечних функцій.
- `<windows.h>`: Використовується для налаштування кодування консолі в Windows.

- "process_file.h": Власний заголовний файл, що містить оголошення функцій, які використовуються для обробки файлів.

2. Основний цикл програми:

```
while (true) {  
    std::cout << "\n\nОбробка текстових файлів\n";  
    std::cout << "1. Ввести ім'я вхідного файлу\n";  
    std::cout << "2. Завершити програму\n";  
    std::cout << "Виберіть пункт меню: ";  
    int choice;  
    std::cin >> choice;
```

- Програма працює у безкінечному циклі, доки користувач не вибере опцію для завершення програми. Користувачу пропонується вибрати одну з двох опцій: ввести ім'я вхідного файлу або завершити програму.

3. Обробка вибору користувача:

```
if (choice == 1) {  
    char input_filename[100];  
    std::cout << "Введіть ім'я вхідного файлу: ";  
    std::cin >> input_filename;  
    std::ifstream input_file(input_filename, std::ios::in);  
    if (!input_file.is_open()) {  
        std::cerr << "Файл не знайдено: " << input_filename << std::endl;  
        continue;  
    }  
    char* output_filename = nullptr;
```

- Якщо користувач обирає першу опцію, програма запитує ім'я вхідного файлу та перевіряє, чи файл можна відкрити для читання. Якщо файл не знайдено, програма виводить повідомлення про помилку і повертається до основного меню.

4. Меню обробки файлу:

```
while (true) {  
    std::cout << "\n\nОбробка файлу: " << input_filename << "\n";  
    std::cout << "1. Вивести вхідний файл\n";  
    std::cout << "2. Обробити файл\n";  
    std::cout << "3. Вивести кількість цифр у вхідному файлі\n";  
    std::cout << "4. Вивести вихідний файл\n";  
    std::cout << "5. Завершити роботу з файлом\n";  
    std::cout << "Виберіть пункт меню: ";  
    int file_choice;  
    std::cin >> file_choice;
```

- У цьому циклі користувачу пропонується меню для вибору операції над вхідним файлом: вивести вміст файлу, обробити файл, вивести кількість цифр у файлі, вивести вихідний файл або завершити роботу з файлом.

5. Обробка вибору операції над файлом:

- Вивести вхідний файл:

```
if (file_choice == 1) {  
    print_file_content(input_filename);  
}
```

- Обробити файл:

```
else if (file_choice == 2) {  
    if (output_filename) {  
        delete[] output_filename;  
    }  
    output_filename = generate_output_filename(input_filename);  
    if (!output_filename) {  
        std::cerr << "Не вдалося отримати ім'я вихідного файлу.\n";
```



```

        continue;
    }

    process_file(input_filename, output_filename);
}

- Вивести кількість цифр у вхідному файлі:

    else if (file_choice == 3) {

        std::cout << "Кількість цифр у вхідному файлі: " <<
count_digits(input_filename) << std::endl;

    }

- Вивести вихідний файл:

    else if (file_choice == 4) {

        if (!output_filename) {

            std::cerr << "Спочатку обробіть файл.\n";

            continue;

        }

        print_file_content(output_filename);

    }

- Завершити роботу з файлом:

    else if (file_choice == 5) {

        break;

    }

- Некоректний вибір:

    else {

        std::cerr << "Неправильний вибір пункту меню!\n";

    }

```

6. Очищення пам'яті для вихідного файлу:

```
if (output_filename) {  
    delete[] output_filename;  
}
```

7. Завершення програми:

```
else if (choice == 2) {  
    std::cout << "Програма завершена.\n";  
    break;  
}  
else {  
    std::cerr << "Неправильний вибір пункту меню!\n";  
}
```

- Якщо користувач обирає завершити програму, цикл завершується і програма виводить повідомлення про завершення.

Функції, що використовуються в програмі :

- ``print_file_content``: Функція для виведення вмісту файлу.
- ``process_file``: Функція для обробки файлу, наприклад, копіювання або модифікація вмісту.
- ``count_digits``: Функція для підрахунку кількості цифр у файлі.
- ``generate_output_filename``: Функція для генерації імені вихідного файлу на основі імені вхідного.

Ці функції оголошені у заголовному файлі ``process_file.h`` і реалізовані у відповідному `сpp`-файлі. Вони забезпечують основну функціональність для обробки текстових файлів у програмі.

3. Вихідний текст програми розв'язку задачі (основні фрагменти з коментарями)

1 завдання

```
#include <iostream>

#include <cstring>

#include <windows.h>

// Функція для порівняння двох масивів символів
bool areEqual(char arr1[], char arr2[], int n) {
    for (int i = 0; i < n; ++i) {
        if (arr1[i] != arr2[i]) {
            return false;
        }
    }
    return true;
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int MAX_WORDS = 100;
    char words[MAX_WORDS][20];
    int n = 0; // Лічильник кількості введених слів

    std::cout << "Введіть слова (введіть '+', коли закінчите):\n";
    while (true) {
        std::cin >> words[n];
        if (strcmp(words[n], "+") == 0) {
            break; // Виходимо з циклу, якщо введено 'готово'
        }
    }
```

```
++n;

if (n >= MAX_WORDS) {
    std::cout << "Досягнуто максимальної кількості слів. Завершення
введення.\n";
    break;
}
}
```

// Перебір кожного слова в масиві

```
for (int i = 0; i < n; ++i) {
    // Перевірка чи слово вже було пораховане
    bool isCounted = false;
    for (int j = 0; j < i; ++j) {
        if (areEqual(words[i], words[j], 20)) {
            isCounted = true;
            break;
        }
    }
    if (!isCounted) {
        // Лічильник кількості повторень поточного слова
        int wordCount = 1;
        // Перебір інших слів, щоб знайти входження поточного слова
        for (int k = i + 1; k < n; ++k) {
            if (areEqual(words[i], words[k], 20)) {
                ++wordCount;
            }
        }
    }
}
```

```

    }

    // Вивід слова та кількості його повторень
    std::cout << "Слово: " << words[i] << " Кількість: " << wordCount <<
std::endl;

    }

}

system("pause");

return 0;

}

```

2 завдання

```

#include <iostream>

#include <cstring>

#include <windows.h>

using namespace std;

// Функція, що імітує дії стандартної функції strrchr()
char* my_strrchr(const char* str, char ch) {
    char* lastOccurrence = nullptr;
    while (*str) {
        if (*str == ch) {
            lastOccurrence = const_cast<char*>(str);
        }
        str++;
    }
    return lastOccurrence;
}

```

```
}  
  
// Функція, що імітує дії стандартної функції strncmp()  
int my_strncmp(const char* str1, const char* str2, int n) {  
    int cn = 0;  
    while (*str1 && *str2 && cn < n) {  
        if (*str1 != *str2) {  
            return *str1 - *str2;  
        }  
        str1++;  
        str2++;  
        cn++;  
    }  
  
    // Якщо досягли межі порівняння n або обидва рядки закінчилися  
    одночасно  
    if (cn == n) {  
        return 0;  
    }  
  
    // Якщо один з рядків закінчився раніше  
    if (*str1) {  
        return 1;  
    }  
    else if (*str2) {  
        return -1;  
    }  
}
```

```
else {  
return 0;  
}  
}
```

```
#include <iostream>  
  
int main() {  
SetConsoleCP(1251);  
SetConsoleOutputCP(1251);  
  
int choice;  
  
do {  
cout << "\nМеню:\n";  
  
cout << "1. Знайти останнє входження символу у рядку (strchr)\n";  
cout << "2. Порівняти два рядки (strcmp)\n";  
cout << "3. Вихід\n";  
  
cout << "Введіть свій вибір: ";  
  
cin >> choice;  
  
switch (choice) {  
case 1: {  
const int SIZE = 100;  
char str[SIZE], ch;  
  
cout << "Введіть рядок: ";  
  
cin.ignore();  
cin.getline(str, SIZE);  
  
cout << "Введіть символ, який потрібно знайти: ";  
cin >> ch;
```

```
char* result = my_strchr(str, ch);  
cout << "\nРезультат роботи функції my_strchr(): \n";  
if (result != nullptr) {  
    cout << "Позиція останнього входження символу '" << ch << "' у рядку: "  
    << result - str << endl;  
}  
else {  
    cout << "'" << ch << "' не знайдено у рядку." << endl;  
}
```

```
char* c_result = strchr(str, ch);  
cout << "\nРезультат роботи функції strchr(): \n";  
if (c_result != nullptr) {  
    cout << "Позиція останнього входження символу '" << ch << "' у рядку: "  
    << c_result - str << endl;  
}  
else {  
    cout << "'" << ch << "' не знайдено у рядку." << endl;  
}
```

```
cout << endl;
```

```
if (result == c_result) {  
    cout << "Функції strchr() і my_strchr() повернули однакові значення." <<  
    endl;  
}  
else {
```



```
cout << "Функції strchr() і my_strchr() повернули різні значення." << endl;  
}
```

```
break;
```

```
}
```

```
case 2: {
```

```
char str1[100], str2[100];
```

```
int n;
```

```
cout << "Введіть перший рядок: ";
```

```
cin.ignore();
```

```
cin.getline(str1, 100);
```

```
cout << "Введіть другий рядок: ";
```

```
cin.getline(str2, 100);
```

```
cout << "Введіть кількість символів, які потрібно порівняти: ";
```

```
cin >> n;
```

```
int result = my_strncmp(str1, str2, n);
```

```
cout << "\nРезультат роботи функції my_strncmp(): \n";
```

```
if (result < 0)
```

```
cout << "Перший рядок менший за другий." << endl;
```

```
else if (result > 0)
```

```
cout << "Перший рядок більший за другий." << endl;
```

```
else
```

```
cout << "Рядки рівні." << endl;
```

```
int c_result = strncmp(str1, str2, n);
```

```
cout << "\nРезультат роботи функції strncmp(): \n";
```

```
if (c_result < 0)
```

```
cout << "Перший рядок менший за другий." << endl;
```

```
else if (c_result > 0)
```

```
cout << "Перший рядок більший за другий." << endl;
```

```
else
```

```
cout << "Рядки рівні." << endl;
```

```
cout << endl;
```

```
if (result == c_result) {
```

```
cout << "Функції strncmp() і my_strncmp() повернули однакові значення." << endl;
```

```
}
```

```
else {
```

```
cout << "Функції strncmp() і my_strncmp() повернули різні значення." << endl;
```

```
}
```

```
break;
```

```
}
```

```
case 3:
```

```
cout << "Завершення програми." << endl;
```

```
break;
```

default:

```
cout << "Некоректний вибір. Будь ласка, спробуйте ще раз." << endl;
}
} while (choice != 3);
system("pause");
return 0;
}
```

3 завдання

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <cstdlib>
#include <windows.h>
#include "process_file.h"
```

```
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
```

```
while (true) {
    std::cout << "\n\nОбробка текстових файлів\n";
    std::cout << "1. Ввести ім'я вхідного файлу\n";
    std::cout << "2. Завершити програму\n";
    std::cout << "Виберіть пункт меню: ";
```

```
int choice;

std::cin >> choice;

if (choice == 1) {
    char input_filename[100];
    std::cout << "Введіть ім'я вхідного файлу: ";
    std::cin >> input_filename;

    std::ifstream input_file(input_filename, std::ios::in);
    if (!input_file.is_open()) {
        std::cerr << "Файл не знайдено: " << input_filename << std::endl;
        continue;
    }

    char* output_filename = nullptr;

    while (true) {
        std::cout << "\n\nОбробка файлу: " << input_filename << "\n";
        std::cout << "1. Вивести вхідний файл\n";
        std::cout << "2. Обробити файл\n";
        std::cout << "3. Вивести кількість цифр у вхідному файлі\n";
        std::cout << "4. Вивести вихідний файл\n";
        std::cout << "5. Завершити роботу з файлом\n";
        std::cout << "Виберіть пункт меню: ";
```

```
int file_choice;

std::cin >> file_choice;

if (file_choice == 1) {
    print_file_content(input_filename);
}
else if (file_choice == 2) {
    if (output_filename) {
        delete[] output_filename;
    }
    output_filename = generate_output_filename(input_filename);
    if (!output_filename) {
        std::cerr << "Не вдалося отримати ім'я вихідного файлу.\n";
        continue;
    }

    process_file(input_filename, output_filename);
}
else if (file_choice == 3) {
    std::cout << "Кількість цифр у вхідному файлі: " <<
count_digits(input_filename) << std::endl;
}
else if (file_choice == 4) {
    if (!output_filename) {
        std::cerr << "Спочатку обробіть файл.\n";
        continue;
    }
}
```

```

        }

        print_file_content(output_filename);
    }

    else if (file_choice == 5) {
        break;
    }

    else {
        std::cerr << "Неправильний вибір пункту меню!\n";
    }
}

if (output_filename) {
    delete[] output_filename;
}

else if (choice == 2) {
    std::cout << "Програма завершена.\n";
    break;
}

else {
    std::cerr << "Неправильний вибір пункту меню!\n";
}

return 0;
}

```

4. Опис інтерфейсу програми (керівництво користувача)

1 завдання

Ця програма призначена для введення послідовності слів та визначення кількості їх повторень у введеній послідовності.

Інструкції користувача :

1. Запустіть програму.
2. Введіть слова одне за одним у вікно програми.
3. Після кожного введеного слова натисніть клавішу "Enter".
4. Щоб завершити введення слів та отримати результат, введіть символ "+" та натисніть клавішу "Enter".
5. Після цього програма виведе кількість повторень кожного унікального слова у введеній послідовності.

Ця програма допоможе вам легко підрахувати кількість повторень слів у введеному тексті.

2 завдання

Опція 1: Знайти останнє входження символу у рядку (strrchr)

1. Виберіть опцію 1.
2. Введіть рядок, у якому потрібно знайти символ.
3. Введіть символ, який потрібно знайти.
4. Програма покаже результат роботи функції `my_strchr()`, яка знаходить останнє входження символу в рядок.
5. Також буде показано результат роботи стандартної функції `strchr()` для порівняння.
6. Програма порівняє результати обох функцій і виведе, чи вони співпадають.

Опція 2: Порівняти два рядки (strncmp)

7. Виберіть опцію 2.
8. Введіть перший рядок.
9. Введіть другий рядок.
10. Введіть кількість символів, які потрібно порівняти.
11. Програма покаже результат роботи функції `my_strncmp()`, яка порівнює два рядки.
12. Також буде показано результат роботи стандартної функції `strncmp()` для порівняння.
13. Програма порівняє результати обох функцій і виведе, чи вони співпадають.

Опція 3: Вихід

14. Виберіть опцію 3 для завершення роботи програми.

**Якщо введено некоректний вибір, програма виведе повідомлення про помилку і повернеться до головного меню:*

Некоректний вибір. Будь ласка, спробуйте ще раз.

3 завдання

Програма дозволяє працювати з текстовими файлами: виводити вміст файлу, обробляти його, рахувати кількість цифр у файлі та виводити результати в окремий файл.

1. Запустіть програму через файл `.exe`. Відкриється консольне вікно з головним меню.

Головне меню

1. Ввести ім'я вхідного файлу
2. Завершити програму

Введіть номер опції та натисніть Enter.

Обробка файлу

Після введення імені вхідного файлу доступні наступні опції:

1. Вивести вхідний файл – Виводить вміст файлу на екран.
2. Обробити файл – Створює вихідний файл з обробленим вмістом.
3. Вивести кількість цифр у вхідному файлі – Виводить кількість цифр у файлі.
4. Вивести вихідний файл – Виводить вміст вихідного файлу на екран.
5. Завершити роботу з файлом – Повертає до головного меню.

Завершення роботи

- Виберіть "Завершити програму" у головному меню для виходу з програми.

5.Опис тестових прикладів

1 завдання

```
Введіть слова (введіть '+', коли закінчите):
apple
banana
cherry
banana
apple
apple
+
Слово: apple Кількість: 3
Слово: banana Кількість: 2
Слово: cherry Кількість: 1
Press any key to continue
```

```
Введіть слова (введіть '+', коли закінчите):
вводжу слова консоль вводжу текст слова
+
Слово: вводжу Кількість: 2
Слово: слова Кількість: 2
Слово: консоль Кількість: 1
Слово: текст Кількість: 1
Press any key to continue . . .

C:\Users\Slim\Desktop\лаб 8\ConsoleApplication1\ConsoleApplication1.exe
with code 0.
```

2 завдання

```
1. Знайти останнє входження символу у рядку (strrchr)
2. Порівняти два рядки (strcmp)
3. Вихід
Введіть свій вибір: 1
Введіть рядок: рядок
Введіть символ, який потрібно знайти: 0

Результат роботи функції my_strrchr():
'0' не знайдено у рядку.

Результат роботи функції strrchr():
'0' не знайдено у рядку.

Функції strrchr() і my_strrchr() повернули однакові значення.
```

```
Меню:
1. Знайти останнє входження символу у рядку (strrchr)
2. Порівняти два рядки (strcmp)
3. Вихід
Введіть свій вибір: 1
Введіть рядок: рядок
Введіть символ, який потрібно знайти: к

Результат роботи функції my_strrchr():
Позиція останнього входження символу 'к' у рядку: 4

Результат роботи функції strrchr():
Позиція останнього входження символу 'к' у рядку: 4

Функції strrchr() і my_strrchr() повернули однакові значення.
```

```
Меню:
1. Знайти останнє входження символу у рядку (strrchr)
2. Порівняти два рядки (strcmp)
3. Вихід
Введіть свій вибір: 1
Введіть рядок: програмування
Введіть символ, який потрібно знайти: г

Результат роботи функції my_strrchr():
Позиція останнього входження символу 'г' у рядку: 3

Результат роботи функції strrchr():
Позиція останнього входження символу 'г' у рядку: 3

Функції strrchr() і my_strrchr() повернули однакові значення.
```

```
Меню:
1. Знайти останнє входження символу у рядку (strrchr)
2. Порівняти два рядки (strncmp)
3. Вихід
Введіть свій вибір: 2
Введіть перший рядок: test
Введіть другий рядок: teest
Введіть кількість символів, які потрібно порівняти: 2

Результат роботи функції my_strncmp():
Рядки рівні.

Результат роботи функції strncmp():
Рядки рівні.

Функції strncmp() і my_strncmp() повернули однакові значення.
```

```
Меню:
1. Знайти останнє входження символу у рядку (strrchr)
2. Порівняти два рядки (strncmp)
3. Вихід
Введіть свій вибір: 3
Завершення програми.
Press any key to continue . . .
```

3 завдання

```
Введіть назву вхідного файлу: input.txt
Файл успішно оброблено. Вивід записаний inp_out.dat
Ви бажаєте відобразити вміст вхідного файлу? (y/n): y
1 digit
2 digit
3 digit
4 digit
5 digit
6 digit
7 digit
8 digit
9 digit
0
Ви хочете відобразити вміст вихідного файлу? (y/n): y
one digit
two digit
three digit
four digit
five digit
six digit
seven digit
eight digit
nine digit
zero
Total number of digits: 10

C:\Users\Slim\Desktop\лаб 8\ConsoleApplication1\ConsoleApp
e (process 85000) exited with code 0.
Press any key to close this window . . .
```

```
Введіть назву вхідного файлу: input1.txt
Файл успішно оброблено. Вивід записаний inr_out.dat
Ви бажаєте відобразити вміст вхідного файлу? (y/n): y
1 digit
2 digit
3 digit
4 digit
5 digit
0
Ви хочете відобразити вміст вихідного файлу? (y/n): y
one digit
two digit
three digit
four digit
five digit
zero
Total number of digits: 6

C:\Users\Slim\Desktop\лаб 8\ConsoleApplication1\Console
e (process 86280) exited with code 0.
Press any key to close this window . . .
```

```
Обробка текстових файлів
1. Ввести ім'я вхідного файлу
2. Завершити програму
Виберіть пункт меню: 1
Введіть ім'я вхідного файлу: inputttt
Файл не знайдено: inputttt
```

Обробка текстових файлів

1. Ввести ім'я вхідного файлу

2. Завершити програму

Виберіть пункт меню: 1

Введіть ім'я вхідного файлу: input.dat

Обробка файлу: input.dat

1. Вивести вхідний файл

2. Обробити файл

3. Вивести кількість цифр у вхідному файлі

4. Вивести вихідний файл

5. Завершити роботу з файлом

Виберіть пункт меню: 2

Обробка файлу: input.dat

1. Вивести вхідний файл

2. Обробити файл

3. Вивести кількість цифр у вхідному файлі

4. Вивести вихідний файл

5. Завершити роботу з файлом

Виберіть пункт меню: 3

Кількість цифр у вхідному файлі: 6

```
Обробка файлу: input.dat
1. Вивести вхідний файл
2. Обробити файл
3. Вивести кількість цифр у вхідному файлі
4. Вивести вихідний файл
5. Завершити роботу з файлом
Виберіть пункт меню: 4
hello woroned iteight tetwotone anineaonen
```

```
Обробка файлу: input.dat
1. Вивести вхідний файл
2. Обробити файл
3. Вивести кількість цифр у вхідному файлі
4. Вивести вихідний файл
5. Завершити роботу з файлом
Виберіть пункт меню: 5
```

6. Аналіз помилок (опис усунення зауважень)

У даному коді програми помилок не виявлено.

7. Висновки

На цій лабораторній роботі ми вивчили різні аспекти обробки текстової інформації, а також роботи з файлами у мові програмування C++.

У першому завданні ознайомились з обробкою текстової інформації, представленій у масиві символів. Вивела всі слова, які повторюються у введеному тексті, та вказала їх кількість повторень.

У другому завданні дослідили рядкові функції та їх імітацію. Зокрема, реалізувала функції `strchr()` та `strncmp()`, які здійснюють дії стандартних рядкових функцій.

У третьому завданні поглибилися у роботу з файлами. Спочатку створила вхідний файл із текстом, дотримуючись встановлених умов. Потім реалізувала програму, яка читає інформацію з вхідного файлу, виконує певні дії (згідно індивідуального завдання), та записує результати у вихідний файл. Важливим аспектом було забезпечення коректного введення та виведення інформації, а також організація програми у вигляді функцій та розміщення їх у відповідних файлах.

Загалом, ця лабораторна робота дала нам можливість отримати практичні навички з обробки текстової інформації та роботи з файлами у мові програмування C++.