

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 введеним 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 код призначений для підрахунку кількості повторень кожного слова у введеному користувачем тексті і виведення тільки тих слів, які зустрічаються більше одного разу.

Ось опис кожної частини коду:

1. `'countWordOccurrences'`: Ця функція приймає рядок `'text'` і розбиває його на слова, обчислюючи кількість входжень кожного слова у тексті. Результат

зберігається у словнику, де ключами є слова, а значеннями - їх кількість входжень.

2. ``main``: Основна функція програми, яка зчитує введений користувачем текст, викликає функцію ``countWordOccurrences`` для підрахунку кількості входжень слів у тексті, а потім виводить тільки ті слова, які зустрічаються більше одного разу.

Цей код дозволяє ефективно визначати повторюваність слів у введеному тексті та виводити лише ті слова, які зустрічаються більше одного разу, що може бути корисним при аналізі тексту чи підготовці даних для подальшої обробки.

2 завдання

Цей код містить дві функції для роботи з рядками: ``my_strchr`` і ``my_strcmp``, які імітують стандартні функції ``strchr`` і ``strcmp`` відповідно.

1. Функція ``my_strchr`` призначена для пошуку останнього входження певного символу у рядок. Вона проходить по всім символам рядка та зберігає вказівник на останнє входження цього символу. Після проходження всього рядка, функція повертає вказівник на останнє входження або ``nullptr``, якщо символ не знайдено.

2. Функція ``my_strcmp`` призначена для порівняння двох рядків. Вона порівнює кожен символ двох рядків, доки не знайде різницю або досягне кінця одного з рядків. При знаходженні відмінності функція повертає різницю між ASCII-кодами перших неспівпадаючих символів. Якщо рядки ідентичні, повертається 0.

У головній функції ``main`` використовується меню для вибору операції:

- При виборі 1: користувач вводить рядок та символ для пошуку, після чого викликається функція ``my_strchr``, яка виводить останнє входження символу у рядок або повідомлення про відсутність символу.

- При виборі 2: користувач вводить два рядки для порівняння, після чого викликається функція ``my_strcmp``, яка виводить результат порівняння рядків.

- При виборі 3: програма завершує роботу.

Цикл `'do-while'` виконується, поки користувач не обере вихід (вибір 3).

3 завдання

Цей код - це програма для обробки файлів. Ось детальний розбір кожного кроку:

1. Включення бібліотек : Підключаються необхідні бібліотеки, такі як `'<iostream>'`, `'<string>'`, `'<vector>'`, `"file_processing.h"`, і `'<windows.h>'`.
2. Встановлення кодування консолі : `'SetConsoleCP(1251)'` та `'SetConsoleOutputCP(1251)'` встановлюють кодування консолі на Windows-системі.
3. Оголошення змінних : `'inputFileName'`, `'outputFileName'` та `'lineLengths'` оголошуються для зберігання імені вхідного файлу, імені вихідного файлу та довжини кожного рядка вхідного файлу відповідно.
4. Запит імені вхідного файлу : Користувачеві виводиться повідомлення про введення імені вхідного файлу, і користувач вводить ім'я файлу через консоль.
5. Копіювання вмісту вхідного файлу в вихідний файл : Створюється ім'я вихідного файлу шляхом додавання `"_out.dat"` до перших трьох символів імені вхідного файлу. Функція `'copyFile'` (яка, ймовірно, оголошена у файлі `"file_processing.h"`) копіює вміст вхідного файлу в вихідний файл і підраховує кількість рядків у вхідному файлі, зберігаючи довжину кожного рядка у векторі `'lineLengths'`.
6. Повідомлення про успішне завершення роботи : Після завершення обробки файлу виводиться повідомлення про успішне завершення роботи та ім'я створеного вихідного файлу.
7. Виведення вмісту вхідного файлу : Користувачеві пропонується вивести вміст вхідного файлу на екран. Якщо користувач погоджується (`'y'` або `'Y'`), програма відкриває вхідний файл і виводить його вміст рядок за рядком на консоль.
8. Виведення вмісту вихідного файлу : Аналогічно, користувачеві пропонується вивести вміст вихідного файлу на екран. Якщо користувач

погоджується, програма відкриває вихідний файл і виводить його вміст рядок за рядком на консоль.

9. Завершення програми : Після виведення вмісту файлів програма завершується з кодом виходу 0.

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

1 завдання

```
#include <iostream>
```

```
#include <string>
```

```
#include <map>
```

```
#include <sstream>
```

```
// Функція для розбиття рядка на слова та їх зберігання у словнику
```

```
std::map<std::string, int> countWordOccurrences(const std::string& text) {
```

```
    std::istringstream iss(text);
```

```
    std::string word;
```

```
    std::map<std::string, int> wordCounts;
```

```
    // Читаємо слова з рядка
```

```
    while (iss >> word) {
```

```
        // Збільшуємо лічильник для кожного слова
```

```
        wordCounts[word]++;
```

```
    }
```

```
    return wordCounts;
```

```
}
```

```
int main() {
```

```
    std::string text;
```

```
    // Зчитуємо текст від користувача
```

```
    std::cout << "Введіть текст: ";
```

```
    std::getline(std::cin, text);
```

```
    // Підраховуємо кількість входжень кожного слова у тексті
```

```
    std::map<std::string, int> wordCounts = countWordOccurrences(text);
```

```
    // Виводимо тільки слова, які зустрічаються більше одного разу
```

```
    std::cout << "Слова, які повторюються:\n";
```

```
    for (const auto& pair : wordCounts) {
```

```
        if (pair.second > 1) {
```

```
            std::cout << pair.first << ": " << pair.second << std::endl;
```

```
        }
```

```
    }
```

```
}
```

2 завдання

```
#include <iostream>
```

```
#include <cstring>
```

```
using namespace std;
```

```
// Функція, що імітує дії стандартної функції 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;  
}
```

```
// Функція, що імітує дії стандартної функції strcmp()
```

```
int my_strcmp(const char* str1, const char* str2) {  
    // Порівнюємо символи у рядках  
    while (*str1 && *str2) {  
        if (*str1 != *str2)  
            return *str1 - *str2;  
        str1++;  
        str2++;  
    }  
    return *str1 - *str2;  
}
```

```
#include <iostream>
```

```
int main() {
```

```
    int choice;
```

```
    do {
```

```
        std::cout << "\nMenu:\n";
```

```
        std::cout << "1. strchr\n";
```

```
        std::cout << "2. strncmp\n";
```

```
        std::cout << "3. Exit\n";
```

```
        std::cout << "Enter your choice: ";
```

```
        std::cin >> choice;
```

```
        switch (choice) {
```

```
            case 1: {
```

```
                const int SIZE = 100;
```

```
                char str[SIZE], ch;
```

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

```
                std::cin.ignore();
```

```
                std::cin.getline(str, SIZE);
```

```
                std::cout << "Введіть символ для пошуку: ";
```

```
                std::cin >> ch;
```

```
                char* result = my_strchr(str, ch);
```



```

        if (result != nullptr) {
            std::cout << "Last occurrence of " << ch << " in the string: " <<
result << std::endl;
        }
        else {
            std::cout << "" << ch << " не знайдено в рядку." << std::endl;
        }
        break;
    }
    case 2: {
        char str1[100], str2[100];
        std::cout << "Введіть перший рядок: ";
        std::cin >> str1;
        std::cout << "Введіть другий рядок: ";
        std::cin >> str2;
        int result = my_strcmp(str1, str2);
        if (result < 0)
            std::cout << "Перший рядок менший за другий" << std::endl;
        else if (result > 0)
            std::cout << "Перший рядок більший за другий." << std::endl;
        else
            std::cout << "Рядки рівні." << std::endl;
        break;
    }
    case 3:

```

```

        std::cout << "Вихід з програми." << std::endl;
        break;
    default:
        std::cout << "Невірний вибір. Будь ласка спробуйте ще раз." <<
std::endl;
    }
} while (choice != 3);

return 0;
}

```

3 завдання

```

#include <iostream>
#include <string>
#include <vector>
#include "file_processing.h"
#include <windows.h>

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    std::string inputFileName, outputFileName;
    std::vector<int> lineLengths;

    // Запит ім'я вхідного файлу
    std::cout << "Введіть назву вхідного файлу: ";
    std::cin >> inputFileName;
}

```

```
// Копіювання вмісту вхідного файлу у вихідний файл та підрахунок  
кількості рядків і символів
```

```
outputFileName = inputFileName.substr(0, 3) + "_out.dat";  
copyFile(inputFileName, outputFileName, lineLengths);
```

```
// Виведення повідомлення про успішне завершення роботи  
std::cout << "Файл успішно оброблено. Вивід записаний " <<  
outputFileName << std::endl;
```

```
// Запит користувача на виведення тексту вхідного файлу на екран
```

```
char choice;
```

```
std::cout << "Ви бажаєте відобразити вміст вхідного файлу? (y/n): ";
```

```
std::cin >> choice;
```

```
if (choice == 'y' || choice == 'Y') {
```

```
    std::ifstream inputFile(inputFileName);
```

```
    std::string line;
```

```
    while (std::getline(inputFile, line)) {
```

```
        std::cout << line << std::endl;
```

```
    }
```

```
    inputFile.close();
```

```
}
```

```
// Запит користувача на виведення тексту вихідного файлу на екран
```

```
std::cout << "Ви хочете відобразити вміст вихідного файлу? (y/n): ";
```

```
std::cin >> choice;
```

```
if (choice == 'y' || choice == 'Y') {
```

```
std::ifstream outputFile(outputFileName);  
std::string line;  
while (std::getline(outputFile, line)) {  
    std::cout << line << std::endl;  
}  
outputFile.close();  
}  
  
return 0;  
}
```

4. Опис інтерфейсу програми (керівництво користувача)

1 завдання

Ця програма призначена для аналізу тексту та підрахунку кількості повторень кожного слова.

Введення тексту: після запуску програми вам буде запропоновано ввести текст. Просто введіть ваш текст та натисніть клавішу Enter.

Виведення результату: програма виведе слова, які зустрічаються більше одного разу, разом з їхнім кількістю входжень у тексті.

Інструкції щодо використання:

Впевніться, що текст введений без помилок та розділений пробілами між словами.

Програма не розрізняє регістр слів, тож "Hello" і "hello" будуть вважатися однаковими словами.

Деякі спеціальні символи (наприклад, розділові знаки) можуть впливати на визначення слів. Приклад роботи програми є в 5 пункті звіту, це

керівництво допоможе користувачам краще зрозуміти, як працює програма та як отримати очікувані результати.

2 завдання

Ця програма дозволяє виконувати різні операції над рядками, такі як пошук останнього входження символу у рядок та порівняння двох рядків.

Меню:

1. strchr - Знаходження останнього входження символу у рядок.
2. strcmp - Порівняння двох рядків.
3. Вихід - Завершення роботи програми.

Інструкції щодо введення даних:

1. strchr:

- Введіть рядок, в якому потрібно здійснити пошук.
- Введіть символ, який потрібно знайти.

2. strcmp :

- Введіть перший рядок для порівняння.
- Введіть другий рядок для порівняння.

Приклад роботи програми є в 5 пункті звіту, це допоможе користувачеві краще розуміти, як користуватися програмою і отримувати очікувані результати.

3 завдання

Ця програма призначена для обробки текстових файлів, вона дозволяє копіювати вміст вхідного файлу у вихідний файл та відображати їхній вміст на екрані. Ось керівництво користувача:

1. Початок роботи програми:

- Після запуску програми з'явиться запит на введення назви вхідного файлу. Введіть назву файлу, який ви бажаєте обробити.

2. Обробка файлу:

- Після введення назви вхідного файлу програма автоматично створить вихідний файл з ім'ям, яке починається з перших трьох символів вхідного файлу, додаючи "_out.dat" в кінець.

- Вміст вхідного файлу буде скопійовано у вихідний файл, а також програма підрахує кількість рядків у файлі та кількість символів у кожному рядку.

3. Повідомлення про успішне завершення роботи:

- Після завершення обробки файлу виведеться повідомлення про успішне завершення роботи та ім'я вихідного файлу.

4. Відображення вмісту файлів:

- Після завершення обробки файлу програма запитає, чи бажаєте ви переглянути вміст вихідного файлу. Введіть 'y' або 'Y', якщо так.

- Після цього програма також запитає, чи бажаєте ви переглянути вміст вхідного файлу на екрані. Повторно введіть 'y' або 'Y', якщо так.

5. Завершення роботи:

- Після завершення перегляду файлів програма завершить роботу.

5.Опис тестових прикладів

1 завдання

```
Введіть текст: я виконала перше завдання я  
Слова, які повторюються:  
я: 2
```

```
...Program finished with exit code 0  
Press ENTER to exit console.
```

```
Введіть текст: ця програма написана мовою с++ програма написана ця  
Слова, які повторюються:  
написана: 2  
програма: 2  
ця: 2
```

```
...Program finished with exit code 0
```

2 завдання

```
Menu:
1. strchr
2. strncmp
3. Exit
Enter your choice: 1
Введіть рядок: Hello
Введіть символ для пошуку: 1
'1' не знайдено в рядку.
```

```
Menu:
1. strchr
2. strncmp
3. Exit
Enter your choice: 2
Введіть перший рядок: abc
Введіть другий рядок: abcd
Перший рядок менший за другий
```

```
Menu:
1. strchr
2. strncmp
3. Exit
Enter your choice: 3
Вихід з програми.
```

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 inp_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 . . .
```

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++.