

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 9

з дисципліни «Алгоритмізація та програмування»
на тему "Робота з рядками на C ++"

XAI.301. 174. 319. 2 ЛР

Виконав студент гр. 319

06.03.2025

(підпис, дата)

Дворнік І.П

(П.І.Б.)

Перевірів

_____ к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ роботи з низькорівневими рядками на C++ і документацію до класу `string`, а також алгоритми пошуку в рядку, а також реалізувати обробку рядків на C++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1.

А. Вивчити по документації метод стандартного класу `string` відповідно до варіанту (табл.1 №29. `size_t rfind (const char* s, size_t pos = npos) const;`).

В. Визначити функцію, що виконує ті ж дії, що і вивчений метод класу `string`. Вихідний рядок передати першим параметром (масив символів). Для реалізації методу не використовувати функції обробки рядків зі стандартних бібліотек.

С. Викликати свій метод і метод `string`. *Перед викликом ввести з консолі один рядок і зберегти в масиві символів і змінній типу `string`.

Завдання 2.

А.Описати функцію, що обробляє рядок відповідно до завдання з табл.2(String52. Дано рядок-пропозиція російською мовою. Перетворити рядок так, щоб кожне слово починалося з великої літери. Словом вважати набір символів, що не містить пробілів і обмежений пробілами або початком/кінцем рядка. Слова, які не починаються з букви, не змінювати.). Для реалізації можна використовувати функції обробки рядків зі стандартних бібліотек

В.Описати функцію, яка перевіряє, чи задовольняє рядок умовам завдання.

С.* Створити вихідний текстовий файл, що містить не менше 10 різних рядків.

Д.Використовуючи функції 2.А і 2.В, обробити рядок / * текстовий файл рядок за рядком. Додаткові дані ввести з консолі.

Е. Отриманий результат записати у вихідний файл.

Завдання 3.

Завдання 1-2 реалізувати окремими функціями без параметрів, у функції `main()` організувати меню для багаторазового виконання завдань.

Структурувати проєкт програми: винести заголовки і реалізацію функцій в окремі `.h` та `.cpp` файли.

ВИКОНАННЯ РОБОТИ

Завдання 1

Вирішення задачі №29(табл.1).

Вхідні дані:

Букви, слова, речення – мова українська, російська, англійська

Вихідні дані:

Індекс останнього входження підрядка *s* у рядку *string* – цілі числа.

Алгоритм вирішення

1. Вивід меню вибору завдання.
2. Вводимо 1.
3. Вивід запрошення до вводу основного рядка *string*.
4. Вивід запрошення до вводу підрядка *substring to search for*.
5. Пошук останнього входження підрядка *s* у рядку *string*, починаючи з позиції *pos*, двома різними функціями.
6. Виведення обох результатів для порівняння.

Завдання 2

Вирішення задачі String52(табл.2).

Вхідні дані:

Текстовий файл формату .txt – у файлі від 10 строчок на російській мові з використанням символів

Вихідні дані:

Текстовий файл формату .txt - в якому кожне слово рядка починається з великої літери, якщо слово починається з літери – мова російська з використанням символів.

Помилка читання файлу

Алгоритм вирішення(на папері)

1. Вивід меню вибору завдання.
2. Вводимо 2.
3. Відкриття файлу, та перевірка відкрився чи ні.
4. Зчитування файлу.
5. Заміна слів в рядках якщо вони відповідають умові, інакше ні.
6. Запис рядків у новий файл.
7. Закриття вхідного файлу.

8. Закриття вихідного файлу.
9. Повідомлення що результат знаходиться у вихідному файлі.

Алгоритм вирішення(на практиці)

1. Вивід меню вибору завдання.
2. Вводимо 2.
3. Отримуємо повідомлення про помилку зчитування файлу, або про помилку від системи(див. стор.), кінець.

Лістинг коду вирішення задач №29 та String52 наведено в дод. А(стор. 5-8).

Вміст файлу .txt знаходиться у додатку Q(стор. 9)

Екран роботи програм показаний у дод. Б(стор. 10)

ВИСНОВКИ

Було вивчено функції оголошення завдань `void task();`

Закріплено на практиці налагодження меню.

Виникли труднощі з розумінням як повинен працювати код, налагодженням коду через середовище Visual Studio.

ДОДАТОК А

Лістинг коду програми

```
// === file: main.cpp ===
#include <iostream>
#include <fstream>
#include <string>
#include "string_utils.h"

using namespace std;

// Оголошення функцій для окремих завдань
void task1();
void task2();

int main() {
    int choice;
    do {
        // Меню вибору задачі користувачем
        cout << "\n=== Menu ===\n";
        cout << "1. Task 1: Simulation string::rfind\n";
        cout << "2. Task 2: String52.\n";
        cout << "0. Exit\n";
        cout << "You choice: ";
        cin >> choice;
        cin.ignore(); // очищення буфера після введення числа

        // Вибір функції на основі введеного номера
        switch (choice) {
            case 1: task1(); break;
            case 2: task2(); break;
            case 0: cout << "Ending program.\n"; break;
            default: cout << "Wrong choice.\n";
        }
    } while (choice != 0); // Повторювати меню, поки не вибрано вихід

    return 0;
}

// Завдання 1: власна реалізація rfind та порівняння з std::string::rfind
void task1() {
    char cstr[256]; // масив символів для введення C-рядка
    string str;     // об'єкт string для стандартного методу
    string pattern; // підрядок, який будемо шукати

    // Ввід основного рядка
    cout << "Enter a string: ";
```

```

cin.getline(cstr, 256);
str = string(cstr); // перетворення C-рядка в string

// Ввід підрядка
cout << "Enter a substring to search for: ";
getline(cin, pattern);

// Пошук з використанням стандартної функції
size_t std_rfind = str.rfind(pattern);
// Пошук з використанням власної реалізації
int my_rfind = my_rfind_func(cstr, pattern.c_str());

// Виведення обох результатів для порівняння
cout << "[std::string::rfind]: " << std_rfind << endl;
cout << "[my_rfind_func]:      " << my_rfind << endl;
}

// Завдання 2: Обробка рядків з файлу, форматування та запис результату
void task2() {
    ifstream infile("input.txt");    // Вхідний файл
    ofstream outfile("output.txt");  // Вихідний файл

    // Перевірка чи відкрився файл для читання
    if (!infile.is_open()) {
        cerr << "Failed to open input.txt\n";
        return;
    }

    string line; // змінна для збереження поточного рядка з файлу

    // Зчитуємо файл рядок за рядком
    while (getline(infile, line)) {
        // Якщо рядок відповідає умовам – формувати
        if (is_valid_line(line)) {
            string cap = capitalize_words(line);
            outfile << cap << endl; // Запис відформатованого рядка
        }
        else {
            outfile << line << endl; // Інакше – записати без змін
        }
    }

    infile.close(); // Закриття вхідного файлу
    outfile.close(); // Закриття вихідного файлу
    cout << " The result is recorded in output.txt\n"; // Повідомлення
                                                    // користувачу
}

// === file: string_utils.cpp ===

```

```

#include "string_utils.h"
#include <cctype>
#include <cstring>

// Реалізація аналогу string::rfind для C-рядків
int my_rfind_func(const char* str, const char* substr) {
    int len_str = strlen(str);
    int len_sub = strlen(substr);

    if (len_sub == 0 || len_sub > len_str) return -1;

    for (int i = len_str - len_sub; i >= 0; --i) {
        bool found = true;
        for (int j = 0; j < len_sub; ++j) {
            if (str[i + j] != substr[j]) {
                found = false;
                break;
            }
        }
        if (found) return i;
    }
    return -1; // не знайдено
}

// Capitalize слів: перша літера слова – велика, інші не змінюємо
std::string capitalize_words(const std::string& input) {
    std::string result = input;
    bool newWord = true;

    for (size_t i = 0; i < result.length(); ++i) {
        if (isspace(result[i])) {
            newWord = true;
        }
        else if (newWord && isalpha((unsigned char)result[i])) {
            result[i] = toupper((unsigned char)result[i]);
            newWord = false;
        }
        else {
            newWord = false;
        }
    }
    return result;
}

// Перевірка на відповідність умові завдання
bool is_valid_line(const std::string& line) {
    if (line.empty()) return false;
    if (isspace(line[0]) || isspace(line[line.length() - 1])) return false;
    return true;
}

```

```
// === file: string_utils.h ===  
#pragma once  
#include <string>  
  
// Завдання 1 – власна реалізація rfind для C-рядків  
// Повертає позицію останнього входження підрядка або -1  
int my_rfind_func(const char* str, const char* substr);  
  
// Завдання 2 – функція для зміни першої літери кожного слова на велику  
std::string capitalize_words(const std::string& input);  
  
// Завдання 2 – перевірка чи рядок задовольняє умови: непорожній і без пробілів  
// на краях  
bool is_valid_line(const std::string& line);
```


ДОДАТОК Q

#звук_тишины: ничего не слышно, но всё говорит.

вчера+завтра=сегодня? спорный вопрос.

~падающие точки... исчезают в пустоте.

!бесполезная_строка_номер_четыре.

луна > солнце? только ночью.

(размышления)/[хаос] = неопределённость.

никто_не_отвечает: абонент вне зоны смысла.

дерево_в_комнате — почему бы и нет?

<всё_это> может быть ошибкой.

скрытые\файлы\мыслей не подлежат чтению.

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```

=== Menu ===
1. Task 1: Simulation string::rfind
2. Task 2: String52.
0. Exit
You choice: 2
Failed to open input.txt

=== Menu ===
1. Task 1: Simulation string::rfind
2. Task 2: String52.
0. Exit
You choice: 1
Enter a string: шина дверь окно чашка
Enter a substring to search for: дверь
[std::string::rfind]: 5
[my_rfind_func]:      5

=== Menu ===
1. Task 1: Simulation string::rfind
2. Task 2: String52.
0. Exit
You choice:

```

```

=== Menu ===
1. Task 1: Simulation string::rfind
2. Task 2: String52.
0. Exit
You choice: 1
Enter a string: бочка стол нога и поехали они далеко
Enter a substring to search for: и
[std::string::rfind]: 28
[my_rfind_func]:      28

=== Menu ===
1. Task 1: Simulation string::rfind
2. Task 2: String52.
0. Exit
You choice: 2

```

Microsoft Visual C++ Runtime Library



Debug Assertion Failed!

Program: C:\Users\Интроверт\source\repos\Лаб.
9\64\Debug\Лаб. 9.exe

File: minkernel\crts\ucrt\src\appcrt\convert\isctype.cpp
Line: 36

Expression: c >= -1 && c <= 255

For information on how your program can cause an assertion failure, see the Visual C++ documentation on asserts.

(Press Retry to debug the application)

Прервать

Повтор

Пропустить