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 «Реалізація алгоритмів обробки одновимірних масивів мовою С ++

»

ХАІ.301. 175. 318.14 ЛР

Виконав студент гр. \_\_\_\_\_\_318\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_ Катерина ОВЧАРЕНКО*\_\_\_\_\_\_\_*

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_ к.т.н., доц. Олена  ГАВРИЛЕНКО

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ роботи з низькорівневими рядками

на С++ 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.

A. Вивчити по документації метод стандартного класу string відповідно до

варіанту 46

B. 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.

C. Викликати свій метод і метод string аналогічно прикладам коду,

наведеними в дод.А. \*Перед викликом ввести з консолі один рядок і

зберегти в масиві символів і змінній типу string.

Завдання 2.

A.Описати функцію, що обробляє рядок відповідно до завдання 58.

Для реалізації можна використовувати функції обробки рядків зі

стандартних бібліотек

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

C.\* Створити вихідний текстовий файл, що містить не менше 10 різних

рядків.

D.Використовуючи функції 2.А і 2.В, обробити рядок / \* текстовий файл

рядок за рядком. Додаткові дані ввести з консолі.

E. Отриманий результат записати у вихідний файл.

# ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі 46

Вхідні дані (ім’я, опис, тип, обмеження):

str - Масив символів, який представляє вихідний рядок.Тип: константний

масив символів

pos - Початкова позиція в рядку str, з якої починається порівняння.

Тип: беззнаковий цілий тип

len - Довжина підрядка, що порівнюється в рядку str.

Тип: беззнаковий цілий тип.

s - Рядок, з яким порівнюється підрядок рядка str.

Тип: константний масив символів

n - Довжина рядка s, яка порівнюється з підрядком рядка str.

Тип: беззнаковий цілий тип

Вихідні дані (ім’я, опис, тип):  
 custom\_compare: Результат порівняння підрядків двох рядків str та s. Тип: int.

Алгоритм вирішення:

1.Перевірка виходу за межі масиву та порівняння кожного символу

підрядка str

з відповідним символом рядка s.

2.Якщо хоча б один символ відрізняється, повертається відповідне

значення (-1, якщо str менше за s, 1 - якщо str більше за s).

3.Якщо всі символи однакові, але довжини рядків різні, повертається -1,

якщо len < n, і 1, якщо len > n.

4.При ідентичних рядках повертається 0.

Лістинг коду вирішення задачі Matrix 36 наведено в додатку А (стор.)

Екран роботи програми показаний на рис. .

Завдання 2.

Вирішення задачі Array 58

Вхідні дані (ім’я, опис, тип, обмеження):

fullPath - Рядок, що представляє повний шлях до файлу.

Тип: посилання на константний об'єкт рядка

Вихідні дані (ім’я, опис, тип):

fileNameWithoutExtension - Рядок, що представляє ім'я файлу без розширення. Тип: рядок що містить ім'я файлу без розширення.

Алгоритм вирішення:

1.Знаходження останнього слеша в рядку fullPath, який вказує на початок

ім'я файлу.

2.Визначення позиції, з якої починається ім'я файлу (startPos), яка дорівнює

або 0 (якщо слеш не знайдено) або наступному символу після останнього

слеша.

3.Знаходження останньої крапки в рядку fullPath, що вказує на початок

розширення файлу.

4.Якщо крапка знайдена та вона знаходиться після слеша і різниця між

позиціями крапки та слеша більше нуля, то повертається підрядок від

startPos до позиції перед крапкою.

5.Якщо крапка не знайдена, або вона знаходиться перед або на позиції

слеша, повертається підрядок від startPos до кінця рядка.

6.Повернення отриманого імені файлу без розширення як результат функції

# ВИСНОВКИ

Шляхом використання функції з введеним користувачем повним шляхом до файлу, було отримано лише ім'я файлу без розширення, демонструючи практичне застосування в програмуванні. Результати роботи дозволили поглибити навички роботи зі строками та розширити розуміння роботи методів рядків у стандартній бібліотеці C++.

ДОДАТОК А

Лістинг коду програми

*#include <iostream>*

*#include <string>*

*int custom\_compare(const char\* str, size\_t pos, size\_t len, const char\* s, size\_t n) {*

*// Перевірка виходу за межі масиву*

*for (size\_t i = 0; i < len && i < n; ++i) {*

*if (str[pos + i] != s[i]) {*

*return (str[pos + i] < s[i]) ? -1 : 1;*

*}*

*}*

*if (len == n) {*

*return 0;*

*} else {*

*return (len < n) ? -1 : 1;*

*}*

*}*

*int main() {*

*// Введення рядка з консолі*

*std::string input\_str;*

*std::cout << "Введіть рядок: ";*

*std::getline(std::cin, input\_str);*

*// Копіювання введеного рядка в масив символів*

*size\_t str\_len = input\_str.length();*

*char\* input\_cstr = new char[str\_len + 1];*

*std::copy(input\_str.begin(), input\_str.end(), input\_cstr);*

*input\_cstr[str\_len] = '\0';*

*// Приклади викликів функції*

*const char\* s = "example";*

*size\_t pos = 2;*

*size\_t len = 3;*

*size\_t n = 3;*

*// Виклик користувацької функції*

*int result\_custom = custom\_compare(input\_cstr, pos, len, s, n);*

*// Виклик методу string::compare*

*std::string input\_str\_sub = input\_str.substr(pos, len);*

*int result\_std = input\_str\_sub.compare(0, len, s, n);*

*// Виведення результатів*

*std::cout << "Результат custom\_compare: " << result\_custom << std::endl;*

*std::cout << "Результат string::compare: " << result\_std << std::endl;*

*// Звільнення виділеної пам'яті*

*delete[] input\_cstr;*

*return 0;*

*}*

*#include <iostream>*

*#include <string>*

*std::string extractFileNameWithoutExtension(const std::string& fullPath) {*

*// Знаходження останнього слеша*

*size\_t lastSlashPos = fullPath.find\_last\_of("/\\");*

*size\_t startPos = (lastSlashPos == std::string::npos) ? 0 : lastSlashPos + 1;*

*// Знаходження останньої крапки після останнього слеша*

*size\_t lastDotPos = fullPath.find\_last\_of('.');*

*// Якщо крапка знайшлася після слеша, виділяємо ім'я файлу без розширення*

*if (lastDotPos != std::string::npos && lastDotPos > startPos) {*

*return fullPath.substr(startPos, lastDotPos - startPos);*

*} else {*

*// Якщо крапки немає, вважаємо весь залишок ім'ям файлу*

*return fullPath.substr(startPos);*

*}*

*}*

*int main() {*

*// Введення повного шляху до файлу*

*std::string fullPath;*

*std::cout << "Введіть повний шлях до файлу: ";*

*std::getline(std::cin, fullPath);*

*// Витягнення імені файлу без розширення*

*std::string fileNameWithoutExtension = extractFileNameWithoutExtension(fullPath);*

*// Виведення результату*

*std::cout << "Ім'я файлу без розширення: " << fileNameWithoutExtension << std::endl;*

*return 0;*

*}*

ДОДАТОК Б

Скрін-шоти вікна виконання програми

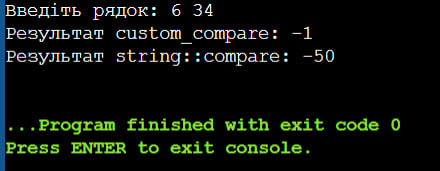


Рисунок Б.1 – Екран виконання програми для вирішення завдання   
46

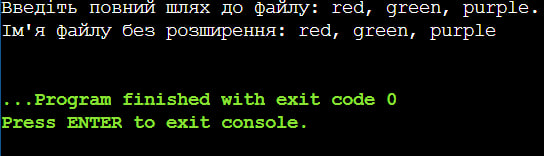


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
58