

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. 141. 319a. 16 ЛР

Виконав студент гр. 319a

Моїсеєнко Євген
(підпис, дата) (П.І.Б.)

Перевірив

 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 (див. табл.1).

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 аналогічно прикладам коду, наведеними в дод.А. *Перед викликом ввести з консолі один рядок і зберегти в масиві символів і змінній типу string.

```
31. size_t rfind (char c, size_t pos = npos) const;
```

Завдання 2.

А.Описати функцію, що обробляє рядок відповідно до завдання з табл.2. 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.

String16. Дано рядок. Перетворити в ньому всі заголовні латинські букви в рядкові.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі 31.

1.А Опис методу rfind класу string

Опис функції

Метод rfind в класі string використовується для пошуку останнього входження символу або підрядка в рядку. Метод починає пошук з кінця рядка і йде в напрямку до початку, повертаючи індекс останнього знайденого входження або `string::npos`, якщо символ або підрядок не знайдено.

Опис параметрів

Метод rfind має декілька перевантажених версій. У випадку пошуку символу, метод має такі параметри:

`charT ch`: символ, який потрібно знайти.

`size_type pos = npos`: початкова позиція для пошуку, якщо не задано, пошук здійснюється з кінця рядка.

Опис результату

Метод повертає `size_type` значення, яке представляє індекс останнього входження символу в рядок. Якщо символ не знайдено, повертається `string::npos`.

1.В Опис власної функції rfind

Опис функції

Функція rfind реалізує функціонал пошуку останнього входження символу в рядку (масиві символів `char`). Функція імітує поведінку методу rfind з класу `string`.

`size_t rfind(const char* str, char c, size_t pos = -1)`

Опис параметрів

`const char* str`: вказівник на початок рядка (масиву символів), в якому буде здійснюватися пошук.

`char c`: символ, який потрібно знайти.

`size_t pos = -1`: початкова позиція для пошуку. За замовчуванням встановлено в -1, що означає початок пошуку з кінця рядка.

Опис результату

Функція повертає `size_t` значення, яке представляє індекс останнього входження символу в рядок. Якщо символ не знайдено, функція повертає -1.

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 8-10).

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі String16.

2.A. Опис функції rfind:

Параметри:

str: Послідовність символів, в якій виконується пошук.

c: Символ, який шукається

pos: Початкова позиція для пошуку (за замовчуванням -1, що означає шукати з кінця рядка).

Результат:

Повертає позицію знайденого символу у рядку або -1, якщо символ не знайдено.

2.B. Лістинг файлу rfind:

```
size_t rfind(const char* str, char c, size_t pos /* = -1 */) {
    size_t len = 0;
    while (str[len] != '\0')
        len++;

    if (pos == -1 || pos > len)
        pos = len;

    if (pos == 0)
        pos = len;

    for (size_t i = pos - 1; i < len; i--) {
        if (str[i] == c)
            return i;
    }
    return -1;
}
```

2.C. Опис функції getResultString:

Ця функція формує рядок з результатами пошуку.

Параметри:

pos: Позиція, знайдена за допомогою методу рядка.

pos_custom: Позиція, знайдена за допомогою власної реалізацією.

Результат:

Рядок з результатами пошуку.

2.D. Лістинг файлу getResultString:

```
string getResultString(size_t pos, size_t pos_custom) {  
    ostringstream oss;  
    oss << "Позиція символу за використанням string: " << pos << endl;  
    oss << "Позиція символу за власною реалізацією: " << pos_custom << endl;  
    return oss.str();  
}
```

Алгоритм вирішення показано на рис. 1

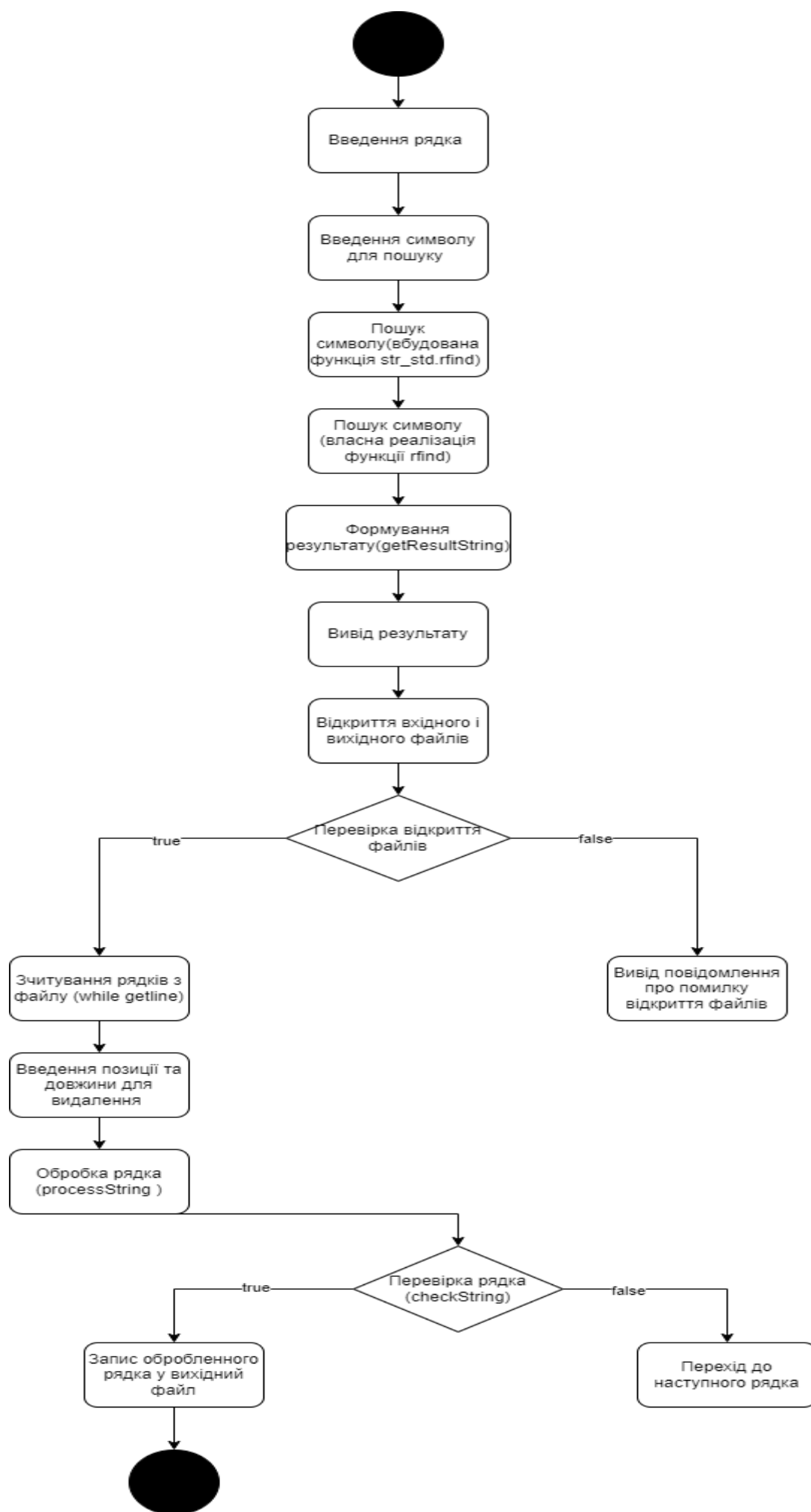


Рисунок 1 – String16.

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 8-10).

Екран роботи програми показаний на рис. Б.2.

ВИСНОВКИ

Вивчення теорії та практична реалізація обробки рядків на C++ у середовищі Visual Studio покращили мої навички програмування та розуміння роботи з рядками, що дозволить мені більш ефективно використовувати їх у майбутніх проектах

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include <sstream>
#include <fstream>
#include <string>

using namespace std;

// Прототипи функцій
size_t rfind(const char* str, char c, size_t pos = -1);
string getResultString(size_t pos, size_t pos_custom);
void processString(string& str, size_t pos, size_t len);
bool checkString(const string& str);

// Реалізація функції rfind
size_t rfind(const char* str, char c, size_t pos /* = -1 */) {
    size_t len = 0;
    while (str[len] != '\0') // Знаходження довжини рядка
        len++;

    if (pos == -1 || pos > len) // Якщо позиція перевищує довжину рядка, змінити
на кінець рядка
        pos = len;

    if (pos == 0) // Якщо позиція рівна нулю, почати пошук з кінця рядка
        pos = len;

    for (size_t i = pos - 1; i < len; i--) { // Прохід по рядку з кінця до
позиції
        if (str[i] == c)
            return i;
    }
    return -1; // Якщо символ не знайдено, повернути -1
}

// Функція для отримання результату у вигляді std::string
string getResultString(size_t pos, size_t pos_custom) {
    ostringstream oss;
    oss << "Позиція символу за використанням string: " << pos << endl;
    oss << "Позиція символу за власною реалізацією: " << pos_custom << endl;
    return oss.str();
}

// Функція обробки рядка
void processString(string& str, size_t pos, size_t len) {

```



```

        str.erase(pos, len);
    }

    // Функція перевірки рядка
    bool checkString(const string& str) {
        // Реалізуйте перевірку умов для рядка
        // Повертає true, якщо рядок відповідає умовам, інакше - false
    }

    int main() {
        const int MAX_SIZE = 100;
        char str[MAX_SIZE];
        string str_std;

        cout << "Введіть рядок: ";
        cin.getline(str, MAX_SIZE); // Зчитати рядок з консолі

        str_std = str; // Конвертувати масив символів в рядок типу string

        char search_char;
        cout << "Введіть символ для пошуку: ";
        cin >> search_char; // Зчитати символ для пошуку

        size_t pos = str_std.rfind(search_char); // Виклик методу rfind для об'єкту
        типу string
        size_t pos_custom = rfind(str, search_char); // Виклик власної реалізацією
        функції

        // Вивести результат у вигляді рядка
        string result = getResultString(pos, pos_custom);
        cout << result;

        ifstream inputFile("C:\\Users\\Жека\\Desktop\\input.txt");
        ofstream outputFile("C:\\Users\\Жека\\Desktop\\output.txt");

        if (!inputFile.is_open() || !outputFile.is_open()) {
            cout << "Помилка відкриття файлів!" << endl;
            return 1;
        }

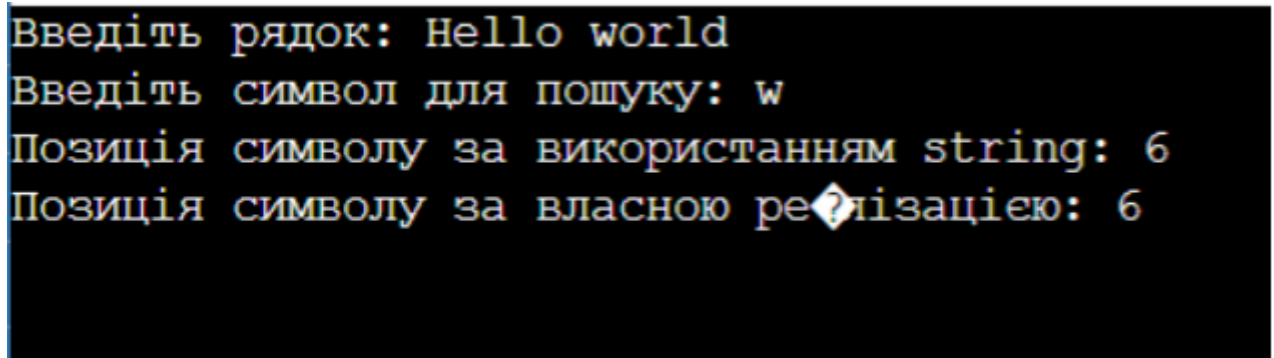
        string line;
        while (getline(inputFile, line)) {
            // Обробка рядка
            size_t pos, len;
            // Введення позиції та довжини видалення з консолі
            cout << "Введіть позицію та довжину видалення для рядка: ";
            cin >> pos >> len;
            processString(line, pos, len);

            // Перевірка рядка

```

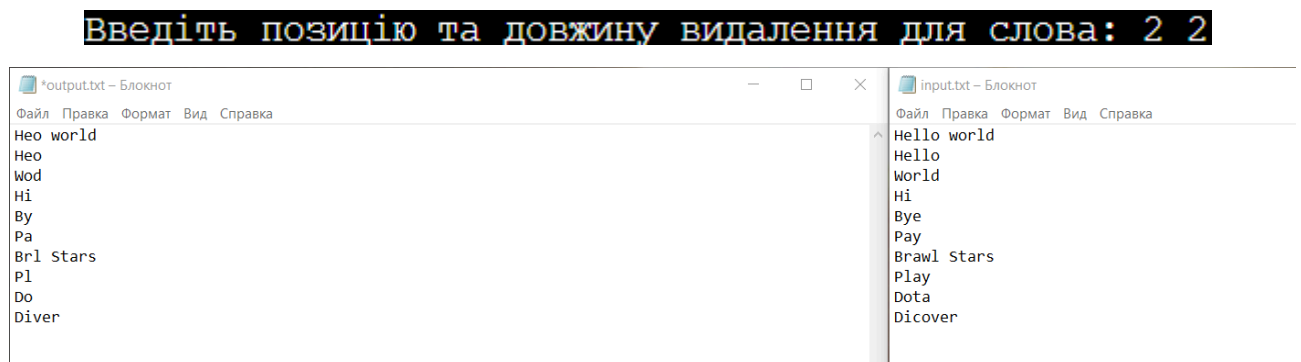
```
        if (checkString(line)) {  
            // Запис обробленого рядка у вихідний файл  
            outputFile << line << endl;  
        }  
    }  
  
    inputFile.close();  
    outputFile.close();  
  
    return 0;  
}
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми



```
Введіть рядок: Hello world
Введіть символ для пошуку: w
Позиція символу за використанням string: 6
Позиція символу за власною реалізацією: 6
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
31.



Введіть позицію та довжину видалення для слова: 2 2

| output.txt – Блокнот | input.txt – Блокнот |
|----------------------|---------------------|
| Heo world | Hello world |
| Heo | Hello |
| Wod | world |
| Hi | Hi |
| By | Bye |
| Pa | Pay |
| Brl Stars | Brawl Stars |
| Pl | Play |
| Do | Dota |
| Diver | Discover |

Рисунок Б.2 – Екран виконання програми для вирішення завдання
String16.