

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. 175. 318. 08 ЛР

Виконав студент гр. _____ 318

_____ Каріна ГЛІБОВА
(підпис, дата) (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 (див. табл.1).

Вивчити по документації метод стандартного класу `string` `string& insert (size_t pos, const string& str)`.

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`.

Завдання 2.

А. Описати структуру, яка містить всі вхідні і всі вихідні дані задачі.

В. Визначити функцію (*метод), що реалізує обробку структури відповідно до задачі.

String34. Дано рядки `S` і `S0`. Видалити з рядка `S` останню підстроку, що збігається з `S0`. Якщо таких підстрок немає, то вивести рядок `S` без змін.

С. Визначити функцію (*метод), що перевіряє на коректність і заповнює відповідні поля вхідних даних структури.

Д. Викликати функції (*методи) з пунктів С, В після оголошення змінної (об'єкту) структури.

Е. Вивести значення полів вихідних даних.

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.

1А.

Метод `string& insert(size_t pos, const string& str)` є функцією класу `string` і призначений для вставки одного рядка в інший. Цей метод дозволяє вставити вміст рядка `str` у поточний рядок, починаючи з позиції `pos`. Індксація позицій у рядку починається з нуля. Якщо вказане значення `pos` перевищує довжину поточного рядка, метод викликає виняток типу `out_of_range`.

Використання цього методу змінює поточний рядок, вставляючи в нього нові символи. Метод повертає посилання на той самий, вже змінений об'єкт рядка, що дає змогу застосовувати ланцюжок викликів.

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

`sstr` – початковий рядок, у який буде додана вставка, рядковий тип даних.

`insertText` – те, що потрібно вставити, рядковий тип даних.

`position` – позиція для вставки, ціле число, дійсний тип.

Вихідні дані (ім'я, опис, тип):

`stringResult` – результат, виведений текст зі вставкою, рядковий тип даних.

1В.

Метод `my_insert(char* dest, int pos, const char* insertStr)` є користувацькою реалізацією для вставки одного С-рядка в інший. Функція працює таким чином: спочатку визначається довжина обох рядків — основного (`dest`) і вставного (`insertStr`). Після цього символи в `dest`, починаючи з позиції `pos`, зміщуються вправо на довжину вставного рядка, щоб звільнити місце для вставки. Потім відбувається копіювання символів з `insertStr` у звільнене місце. У результаті рядок `dest` містить новий текст із вставленим підрядком.

Позиція `pos` повинна бути ненегативною і не перевищувати довжину основного рядка, інакше функція може призводити до помилок.

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

`cstr` – початковий рядок, у який буде додана вставка, масив символів.

`position` – позиція для вставки, ціле число, дійсний тип.

`insertCStr` – те, що потрібно вставити, масив символів.

Вихідні дані (ім'я, опис, тип):

`customResult` – те, що потрібно вставити, масив символів.

Лістинг коду вирішення завдання 1 наведено в дод. А (стор. 7-10).

Екран роботи програми показаний на рис. Б.1. (дод. Б, стор. 11-13)

Приклад діаграми для завдання 1А наведено на рис. Б.4. (дод. Б, стор. 11-13)

Завдання 2.

Вирішення задачі String34.

2А.

Функція `string removeLastSubstring(const string& S, const string& S0)` видаляє останнє входження підрядка `S0` з рядка `S`. Якщо підрядок `S0` не знайдено, повертається початковий рядок без змін.

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

`S` – вхідний рядок, у якому шукається підрядок для видалення, рядковий тип даних.

`S0` – підрядок, останнє входження якого потрібно знайти та видалити з `S`.

Вихідні дані (ім'я, опис, тип):

`result` – виведений рядок без видаленого підрядка, рядковий тип даних.

2В.

Функція `bool hasSubstring(const string& S, const string& S0)` перевіряє, чи міститься підрядок `S0` у рядку `S`. Визначає, чи входить `S0` як частина `S`.

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

`S` – основний рядок, у якому здійснюється пошук, логічний тип.

`S0` – підрядок, наявність якого потрібно перевірити в `S`, логічний тип.

Вихідні дані (ім'я, опис, тип):

`true` — якщо `S0` є підрядком `S`.

`false` — якщо `S0` у `S` не зустрічається.

2С.

Яблуко червоне як серце

Червоне сонце заходить

Сонце світить і гріє

Світить вікно у кімнату

У кімнаті червона лампа

Світло лампи червоне і м'яке

М'яке світло вечора

Небо червоне на заході

Море хвилюється червоне

Малюю червоне коло на папері

2Е.

Яблуко червоне як серце

Червоне сонце заходить

Сонце світить і гріє

Світить вікно у кімнату

У кімнаті

Світло лампи червоне і м'яке

М'яке світло вечора

Небо червоне на заході

Море хвилюється червоне

Малюю червоне коло на папері

Лістинг коду вирішення задачі String34 наведено в дод. А (стор. 7-10).

Екран роботи програми показаний на рис. Б.2. (дод. Б, стор. 11-13)

Приклад діаграм для завдань 2А та 2D наведено на рис. Б.5. (дод. Б, стор. 11-

13)

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ роботи з низькорівневими рядками на C++ і розглянуто документацію до класу `string`. На практиці було опрацьовано алгоритми пошуку в рядку, а також реалізовано обробку рядків на C++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include <string>
#include "windows.h"
#include "insert.h"
#include "string.h"

using namespace std;

int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    int menu; // Зміна для номеру завдання
    do
    { // початок циклу
        cout << "Номер завдання: "; //введення номеру завдання
        cin >> menu; // обирає номеру завдання
        cout << endl; //вільна строка
        switch (menu) {
            case 1: task_1(); break; // 1 - завдання 1
            case 2: task_2(); break; // 2 - завдання 2
            // case 3: task_3(); break; // 3 - завдання 3
            case -1: cout << "Вихід..." << endl; break; // -1 - вихід
            default: cout << "Помилка! Лише 1, 2!" << endl; // інший номер - повторити
        }
        cout << endl; // вільна строка
        cout << "+-----+" << endl; // строка задля полегшення
        візуального сприймання тексту
        cout << endl; // вільна строка
    } // кінець циклу
    while (menu != -1); // умова виконання циклу
    return 0;

#ifdef INSERT_H
#define INSERT_H

void task_1();

#endif // INSERT_H

#include <iostream>
#include <string>
#include "insert.h"
using namespace std;

// В. Реалізація власної функції вставки в масив символів (без функцій string.h)
void my_insert(char* dest, int pos, const char* insertStr) {
    int lenDest = 0, lenInsert = 0;

    // Довжина вихідного рядка
    while (dest[lenDest] != '\0') lenDest++;
    // Довжина вставного рядка
    while (insertStr[lenInsert] != '\0') lenInsert++;

```

```

    // Зсув елементів вправо
    for (int i = lenDest; i >= pos; --i) {
        dest[i + lenInsert] = dest[i];
    }

    // Вставка символів
    for (int i = 0; i < lenInsert; ++i) {
        dest[pos + i] = insertStr[i];
    }
}

// A + C. Виклик методу insert() з класу string та виклик своєї функції
void task_1() {
    char cstr[256];      // Масив символів
    string sstr;         // Об'єкт класу string

    cout << "Введіть рядок: ";
    cin.ignore();
    cin.getline(cstr, 256); // Зчитування в масив символів
    sstr = cstr;           // Присвоєння в об'єкт string

    string insertText;
    cout << "Введіть вставку: ";
    cin.clear();
    cin >> insertText;
    const char* insertCStr = insertText.c_str();

    int position; // позиція для вставки
    cout << "Введіть позицію для вставки: ";
    cin.clear();
    cin >> position;

    // Виклик методу insert з класу string
    string stringResult = sstr;
    stringResult.insert(position, insertText);

    // Виклик власної функції вставки
    char customResult[512]; // буфер для результату
    for (int i = 0; i <= strlen(cstr); ++i) {
        customResult[i] = cstr[i];
    }
    my_insert(customResult, position, insertCStr);

    // Вивід результатів
    cout << "\nРезультат (string.insert): " << stringResult << endl;
    cout << "Результат (my_insert для char[]): " << customResult << endl;
}

#ifdef STRING_H
#define STRING_H

#include <string>

// Основна функція завдання 2
void task_2();

// Допоміжні функції
std::string removeLastSubstring(const std::string& S, const std::string& S0);
bool hasSubstring(const std::string& S, const std::string& S0);

#endif // STRING_H

```



```

#include "string.h" // Мій файл заголовку
#include <iostream> // Для вводу/виводу в консоль
#include <fstream> // Для роботи з файлами
#include <string> // Для використання типу string
#include <vector> // Для використання векторів, щоб перевіряти підрядок
using namespace std; // Щоб не писати std:: перед кожним стандартним елементом

// А. Функція, що видаляє останнє входження підрядка S0 з рядка S
string removeLastSubstring(const string& S, const string& S0) {
    size_t pos = S.rfind(S0); // Знаходимо позицію останнього входження S0 у S
    if (pos != string::npos) { // Якщо входження знайдено
        string result = S; // Створюємо копію рядка S
        result.erase(pos, S0.length()); // Видаляємо підрядок, починаючи з pos
        return result; // Повертаємо результат
    }
    return S; // Якщо входження не знайдено – повертаємо S без змін
}

// Б. Функція, що перевіряє, чи міститься S0 у S
bool hasSubstring(const string& S, const string& S0) {
    return S.find(S0) != string::npos; // Повертає true, якщо S0 є в S
}

// В. Функція для виводу вмісту файлу на консоль
void printFileContent(const string& filename) {
    ifstream file(filename); // Відкриваємо файл для читання
    if (!file.is_open()) { // Якщо файл не відкрився
        cerr << "Не вдалося відкрити файл: " << filename << endl; // Виводимо помилку
        return; // Виходимо з функції
    }

    string line; // Змінна для зберігання рядків з файлу
    cout << "Вміст файлу " << filename << ":\n\n"; // Повідомлення для користувача
    while (getline(file, line)) { // Читаємо файл рядок за рядком
        cout << line << endl; // Виводимо кожен рядок
    }
    file.close(); // Закриваємо файл
}

// Д. Функція для зчитування підрядка з консолі
string readSubstringFromUser() {
    string S0; // Змінна для збереження введеного рядка
    cout << "\nВведіть підрядок для видалення: "; // Запит до користувача
    cin.ignore(); // Очищення буфера вводу (захист від залишків після cin)
    getline(cin, S0); // Зчитуємо рядок повністю (з пробілами)
    return S0; // Повертаємо введений рядок
}

// Е. Функція для обробки рядків: видалення останнього входження S0
void processStrings(const string& inputFile, const string& outputFile, const string& S0) {
    // Відкриваємо вхідний файл для читання
    ifstream inFile(inputFile);
    if (!inFile.is_open()) {
        // Якщо файл не вдалося відкрити, виводимо помилку та виходимо
        cerr << "Не вдалося відкрити вхідний файл!" << endl;
        return;
    }

    vector<string> lines; // Вектор для збереження всіх рядків з файлу
    bool found = false; // Прапорець, що показує, чи знайдено підрядок хоча б
    в одному рядку

```

```

    string line; // Змінна для зчитування поточного рядка
    while (getline(inFile, line)) { // Зчитуємо кожен рядок з файлу
        if (!found && hasSubstring(line, S0)) { // Якщо підрядок ще не знайдено,
перевіряємо цей рядок
            found = true; // Якщо знайдено, встановлюємо
прапорець
        }
        lines.push_back(line); // Додаємо поточний рядок до вектору
    }
    inFile.close(); // Закриваємо вхідний файл після
читання всіх рядків

    if (!found) { // Якщо підрядок не знайдено у жодному рядку
        cout << "\nПідрядок \" " << S0 << "\" не знайдено у жодному рядку. "
            << "Файл залишився без змін.\n" << endl; // Повідомляємо користувача
        return; // Виходимо з функції, не створюючи вихідний файл
    }

    // Відкриваємо вихідний файл для запису
    ofstream outFile(outputFile);
    if (!outFile.is_open()) {
        // Якщо не вдалося відкрити вихідний файл, повідомляємо про помилку
        cerr << "Не вдалося відкрити вихідний файл!" << endl;
        return;
    }

    // Обробляємо кожен збережений рядок
    for (const string& l : lines) {
        if (hasSubstring(l, S0)) {
            // Якщо рядок містить підрядок – видаляємо останнє входження
            string modified = removeLastSubstring(l, S0);
            outFile << modified << endl; // Записуємо змінений рядок у вихідний файл
        }
        else {
            // Інакше записуємо рядок без змін
            outFile << l << endl;
        }
    }
    outFile.close(); // Закриваємо вихідний файл після завершення запису
}

// Головна функція, яка виконує всі кроки
void task_2() {
    const string inputFile = "string_in_8.txt"; // Назва вхідного файлу
    const string outputFile = "string_out_8.txt"; // Назва вихідного файлу

    printFileContent(inputFile); // Виводимо вміст вхідного файлу на екран

    string S0 = readSubstringFromUser(); // Зчитуємо підрядок, який треба видалити

    processStrings(inputFile, outputFile, S0); // Обробляємо рядки – видаляємо
підрядок

    cout << "\nОбробку завершено. Перевірте файл " << outputFile << endl << endl; //
Повідомлення

    printFileContent(outputFile); // Виводимо результат обробки на екран
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
Номер завдання: 1

Введіть рядок: сон світить
Введіть вставку: це
Введіть позицію для вставки: 3

Результат (string.insert):      сонце світить
Результат (my_insert для char[]): сонце світить
```

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

```
Номер завдання: 2

Вміст файлу string_in_8.txt:

Яблуко червоне як серце
Червоне сонце заходить
Сонце світить і гріє
Світить вікно у кімнату
У кімнаті червона лампа
Світло лампи червоне і м'яке
М'яке світло вечора
Небо червоне на заході
Море хвилюється червоне
Малюю червоне коло на папері

Введіть підрядок для видалення: червона лампа

Обробку завершено. Перевірте файл string_out_8.txt

Вміст файлу string_out_8.txt:

Яблуко червоне як серце
Червоне сонце заходить
Сонце світить і гріє
Світить вікно у кімнату
У кімнаті
Світло лампи червоне і м'яке
М'яке світло вечора
Небо червоне на заході
Море хвилюється червоне
Малюю червоне коло на папері
```

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

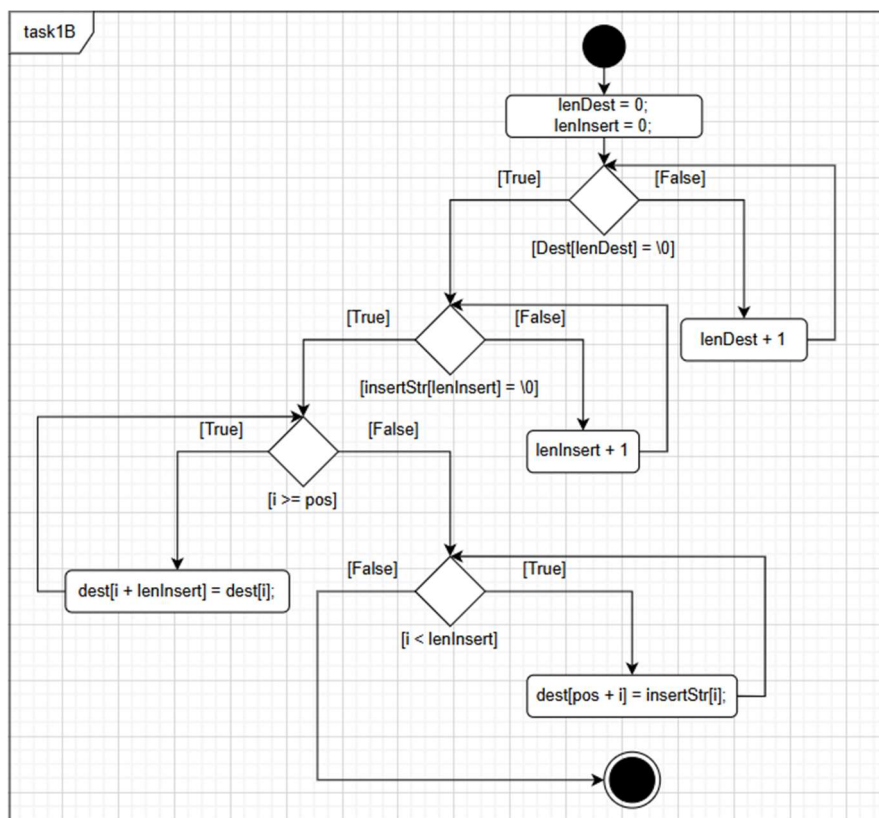


Рисунок Б.3 – Діаграма для завдання 1В

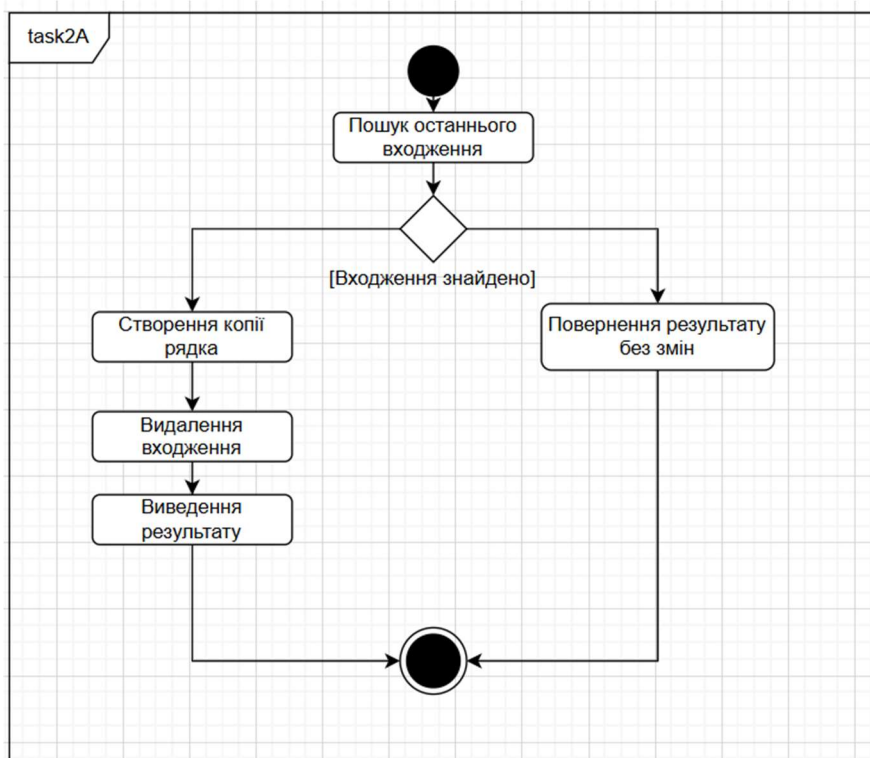


Рисунок Б.4 – Діаграма для завдання 2А

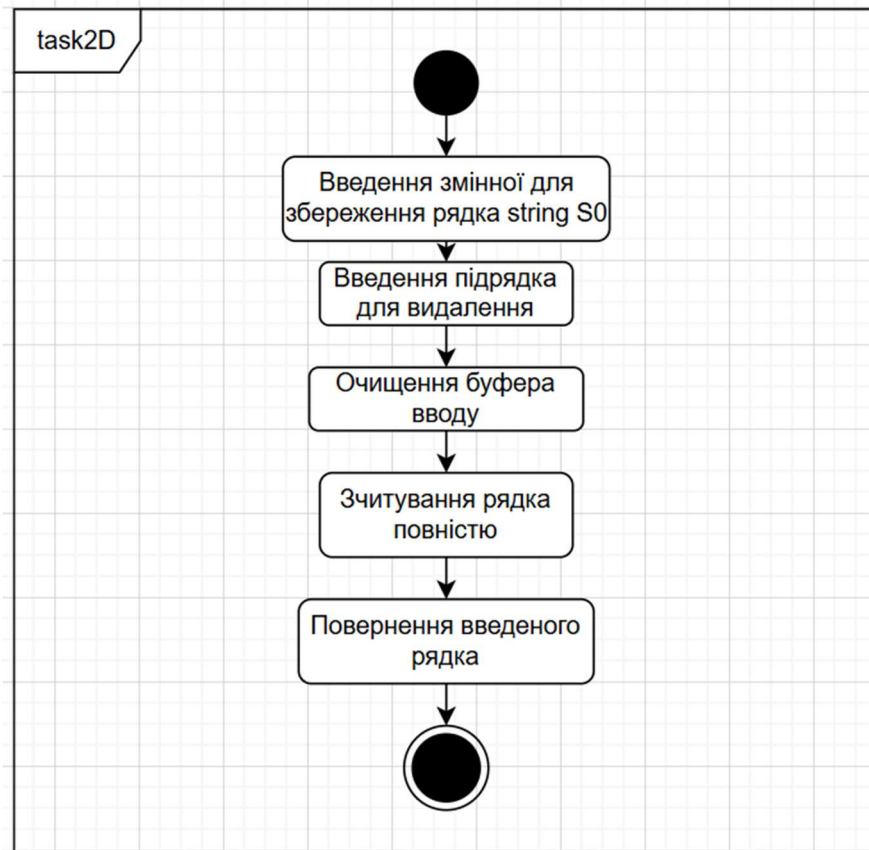


Рисунок Б.5 – Діаграма для завдання 2D