

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»

Тема: "Робота з рядками на C ++"

XAI.301.141.319a. 27 ЛР

Виконав студент гр. 319a

Владислав РУДНЄВ

(Підпис, дата)

(П.І.Б.)

Перевірів к.т.н., доц.

(вчена ступінь, вчене звання)

Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

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.

String_replace19 . string& replace (size_t pos, size_t len, 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 функцію, що обробляє рядок відповідно до завдання з табл.2. Для реалізації можна використовувати функції обробки рядків зі стандартних бібліотек.

String40. Дано рядок, що містить принаймні один символ пробілу. Вивести підстроку, розташовану між першим і останнім пробілом рядка. Якщо рядок містить тільки один пробіл, то вивести порожній рядок.

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 `String_replace19`.

Вхідні дані:

- **str1** — вхідний рядок, який вводиться користувачем.
 - Тип: `std::string`.
 - Обмеження: довжина до 1024 символів.
- **pos** — позиція, з якої починається заміна.
 - Тип: `size_t`.
 - Повинна задовольняти умову: $0 \leq pos < \text{довжина рядка}$.
- **len** — кількість символів, які потрібно замінити.
 - Тип: `size_t`.
 - Повинна задовольняти умову: $0 \leq len \leq \text{довжина рядка} - pos$.
- **replacement** — новий підрядок, який буде вставлено на місце заміни.
 - Тип: `std::string`.
 - Обмеження: довжина до 1024 символів.

Вихідні дані:

- **str1_copy** — модифікований рядок після заміни методом `std::string::replace`.
 - Тип: `std::string`.
- **str2** — модифікований C-style рядок після заміни власною функцією.
 - Тип: `char[]`.

Алгоритм:

1. Користувач вводить рядок `str1`, який зчитується повністю (включаючи пробіли).
2. Рядок копіюється в C-style масив `str2` за допомогою функції `strcpy_s`.
3. Користувач вводить позицію `pos`, довжину `len` і рядок `replacement`.
4. Здійснюється перевірка на коректність значень `pos` і `len`.
5. Виконується заміна підрядка в `str1_sору` за допомогою стандартного методу `replase`.
6. Здійснюється заміна підрядка в `str2` за допомогою самостійно реалізованої функції.
7. Обидва результати виводяться на екран.

Завдання 2.

Вирішення задачі String40.

Вхідні дані:

- **str** — рядок, який вводиться користувачем з клавіатури.
 - Тип: `std::string`. Повинен містити щонайменше два пробіли.
- **input.txt** — текстовий файл, кожен рядок якого містить фразу або речення. Кожен рядок обробляється окремо.

Вихідні дані:

- Підрядок між першим і останнім пробілом — виводиться в консоль, якщо вхідний рядок має принаймні два пробіли.
 - Тип: `std::string`.
- **output.txt** — текстовий файл, в який записуються результати обробки кожного рядка з **input.txt**:
 - Якщо рядок містить два або більше пробіли — у файл записується підрядок між першим і останнім пробілом.
 - Якщо недостатньо пробілів — записується повідомлення `[NO VALID SUBSTRING]`.

Алгоритм:

1. Користувач вводить рядок `str`.
2. Перевіряється, чи містить він хоча б два пробіли за допомогою функції `has_spaces(str)`:
 - Якщо так — викликається `extract_between_spaces(str)` і результат виводиться на екран.
 - Якщо ні — виводиться повідомлення про помилку.
3. Відкривається файл `input.txt` для читання і `output.txt` для запису.
4. Для кожного рядка з `input.txt`:
 - Якщо в ньому є хоча б два пробіли, до `output.txt` записується підрядок між першим і останнім пробілом.
 - Інакше — записується повідомлення `[NO VALID SUBSTRING]`.
5. Файли закриваються. Виводиться повідомлення про успішне завершення обробки.

Лістинг коду дод. А (стр.6 — 7)

Скрін-шоти вікна виконання програми дод. Б (сторінка 8 — 10)

ВИСНОВОК

В процесі виконання завдання було вивчено теоретичні основи роботи з низькорівневими рядками на C++, а також опрацьовано документацію до стандартного класу `std::string`. Ознайомлення з алгоритмами пошуку в рядках дозволило краще зрозуміти механізми маніпуляції текстовими даними. Практична реалізація обробки рядків у середовищі Visual Studio закріпила отримані знання та навички програмування з використанням як низькорівневих C-style рядків, так і об'єктно-орієнтованого підходу з класом `std::string`. Це сприяло формуванню цілісного розуміння роботи з рядками в мові C++ та підвищенню рівня програмістських компетенцій.

ДОДАТОК А

Лістинінг коду

№main.cpp

```
#include <iostream>    // Для вводу/виводу
#include <string>       // Для роботи з std::string
#include <cstring>       // Для роботи з функціями C-рядків (strcpy_s)
#include "String_replace.h" // Заголовок з функціями завдання 1
#include "String40.h"    // Заголовок з функціями завдання 2

using namespace std;

// --- Функція виконання завдання 1 ---
// Демонструє роботу як стандартного методу string::replace, так і власноруч реалізованої функції
void run_task1() {
    string str1;
    cout << "Enter string: ";
    getline(cin, str1); // Зчитуємо весь рядок з пробілами

    // Копіюємо string у C-style масив символів
    char str2[1024]; // Буфер для C-style рядка
    strcpy_s(str2, str1.c_str()); // Безпечне копіювання рядка у масив

    size_t pos, len;
    string replacement;

    // Зчитування параметрів для заміни
    cout << "Enter position to replace: ";
    cin >> pos;
    cout << "Enter length to replace: ";
    cin >> len;
    cout << "Enter replacement text: ";
    cin.ignore(); // Очищаємо буфер після cin
    getline(cin, replacement); // Зчитуємо текст заміни

    // --- Заміна в std::string ---
    string str1_copy = str1; // Робимо копію, щоб не змінювати оригінал
    string_replace_std(str1_copy, pos, len, replacement.c_str());
    cout << "Result using std::string::replace: " << str1_copy << "\n";

    // --- Заміна в C-style рядку ---
    string_replace_custom(str2, pos, len, replacement.c_str());
    cout << "Result using custom replace: " << str2 << "\n";
}

// --- Функція виконання завдання 2 ---
// Працює з підрядками між пробілами та обробляє файл input.txt
```

```

void run_task2() {
    string str;
    cout << "Enter string with spaces: ";
    getline(cin, str);

    // Перевіряємо наявність пробілів і обробляємо
    if (has_spaces(str)) {
        cout << "Substring between first and last space: " << extract_between_spaces(str) << "\n";
    }
    else {
        cout << "Not enough spaces to extract.\n";
    }

    // Обробляємо рядки з файлу та записуємо результат
    process_file_lines("input.txt", "output.txt");
    cout << "File processed. Check output.txt.\n";
}

// --- Точка входу: головне меню ---
int main() {
    int choice;
    do {
        // Меню користувача
        cout << "\nMenu:\n"
            << "1. Task 1 - Replace substring\n"
            << "2. Task 2 - Substring between spaces\n"
            << "0. Exit\n"
            << "Your choice: ";
        cin >> choice;
        cin.ignore(); // Очищаємо буфер після вводу числа

        // Виконуємо обраний пункт меню
        switch (choice) {
            case 1: run_task1(); break;
            case 2: run_task2(); break;
            case 0: cout << "Exiting.\n"; break;
            default: cout << "Invalid choice.\n";
        }

    } while (choice != 0); // Поки не вибрано вихід

    return 0;
}

```

№String_replace.cpp

```

#include "String_replace.h"
#include <cstring> // Для strlen
using namespace std;

```

```
// Заміна частини рядка за допомогою стандартного методу std::string::replace
void string_replace_std(string& str, size_t pos, size_t len, const char* s) {
    str.replace(pos, len, s); // Просто викликаємо метод replace
}

// Власна реалізація заміни частини рядка у char-масиві без string
void string_replace_custom(char* str, size_t pos, size_t len, const char* s) {
    size_t orig_len = strlen(str); // Поточна довжина вхідного рядка
    size_t rep_len = strlen(s);    // Довжина рядка, який вставляємо

    if (pos > orig_len) return;     // Якщо позиція більша за довжину — нічого не робимо

    // Якщо новий фрагмент довший — зсуваємо символи правіше
    if (rep_len > len) {
        for (int i = (int)orig_len; i >= (int)(pos + len); --i) {
            str[i + rep_len - len] = str[i]; // Зсув символів вправо
        }
    }
    // Якщо новий фрагмент коротший — стискаємо рядок
    else if (rep_len < len) {
        for (int i = (int)(pos + len); i <= (int)orig_len; ++i) {
            str[i + rep_len - len] = str[i]; // Зсув символів вліво
        }
    }

    // Копіюємо нові символи на місце заміни
    for (size_t i = 0; i < rep_len; ++i) {
        str[pos + i] = s[i];
    }

    // Символ '\0' автоматично переміщено при зсуві, тож рядок залишається коректним
}
```

№String_replace.h

```
#pragma once // Захищає від подвійного включення
```

```
#ifndef TASK1_H
```

```
#define TASK1_H
```

```
#include <string> // Для std::string
```

```
// Заміна частини рядка std::string на новий текст
```

```
void string_replace_std(std::string& str, size_t pos, size_t len, const char* s);
```

```
// Заміна частини рядка у звичайному char-масиві (без std::string)
```

```
void string_replace_custom(char* str, size_t pos, size_t len, const char* s);
```



```
#endif
```

№String40.cpp

```
#include "String40.h"
#include <fstream> // Для роботи з файлами (ifstream, ofstream)
using namespace std;
// Функція перевіряє, чи рядок має хоча б два пробіли
bool has_spaces(const string& str) {
    // Повертає true, якщо перший і останній пробіли знаходяться в різних місцях
    return str.find(' ') != string::npos && str.rfind(' ') != str.find(' ');
}

// Функція повертає підрядок, який знаходиться між першим і останнім пробілом у рядку
std::string extract_between_spaces(const std::string& str) {
    size_t first = str.find(' ');    // Індекс першого пробілу
    size_t last = str.rfind(' ');    // Індекс останнього пробілу

    // Якщо знайдено два різні пробіли
    if (first != string::npos && last != string::npos && first != last)
        // Повертаємо підрядок між ними
        return str.substr(first + 1, last - first - 1);

    // Якщо пробіл один або взагалі немає — повертаємо порожній рядок
    return "";
}

// Функція читає рядки з вхідного файлу, обробляє кожен рядок і записує результат у вихідний файл
void process_file_lines(const string& inFile, const string& outFile) {
    ifstream fin(inFile);    // Відкриваємо файл для читання
    ofstream fout(outFile);  // Відкриваємо файл для запису
    string line;

    // Зчитуємо файл рядок за рядком
    while (getline(fin, line)) {
        // Якщо в рядку є хоча б два пробіли
        if (has_spaces(line)) {
            // Витягуємо підрядок між пробілами та записуємо у вихідний файл
            fout << extract_between_spaces(line) << "\n";
        }
        else {
            // Якщо недостатньо пробілів — записуємо повідомлення
            fout << "[NO VALID SUBSTRING]" << "\n";
        }
    }
}

// Закриваємо файли
fin.close();
fout.close();
```

```
}
```

№String40.h

```
#pragma once // Захист від повторного включення файлу (альтернатива #ifndef)
```

```
// Початок класичного захисту від багаторазового включення
```

```
#ifndef TASK2_H
```

```
#define TASK2_H
```

```
#include <string> // Підключення бібліотеки для роботи з типом std::string
```

```
using namespace std;
```

```
// --- Перевірка наявності хоча б двох пробілів у рядку ---
```

```
// Повертає true, якщо у рядку є мінімум два пробіли
```

```
bool has_spaces(const std::string& str);
```

```
// --- Повертає підрядок між першим і останнім пробілом у рядку ---
```

```
// Якщо пробіл лише один або жодного — повертає порожній рядок
```

```
string extract_between_spaces(const std::string& str);
```

```
// --- Обробляє текстовий файл пострічково ---
```

```
// Зчитує з файлу `inFile`, для кожного рядка викликає extract_between_spaces()
```

```
// Результати записує у файл `outFile`
```

```
void process_file_lines(const std::string& inFile, const std::string& outFile);
```

```
// Кінець класичного захисту
```

```
#endif
```

№input.txt

```
Hello world this is C++
```

```
Line without space
```

```
Another sample with spaces
```

```
SimpleTest
```

```
123 456 789
```

№output.txt

```
world this is
```

```
without
```

```
sample with
```

```
[NO VALID SUBSTRING]
```

```
456
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми та діаграми

На рис.1 показано виконання коду лабораторної роботи №9

```
Menu:
1. Task 1 - Replace substring
2. Task 2 - Substring between spaces
0. Exit
Your choice: 1
Enter string: HelloWorld
Enter position to replace: 3
Enter length to replace: 5
Enter replacement text: REP
Result using std::string::replace: HelREPld
Result using custom replace: HelREPld

Menu:
1. Task 1 - Replace substring
2. Task 2 - Substring between spaces
0. Exit
Your choice: 2
Enter string with spaces: Hello World
Not enough spaces to extract substring.
File processed. Check output.txt.

Menu:
1. Task 1 - Replace substring
2. Task 2 - Substring between spaces
0. Exit
Your choice: 0
Exiting.
```

Рисунок 1 — перевірка правильності виконання коду

На рис.2 показано діаграму коду програми для завдання 1.В

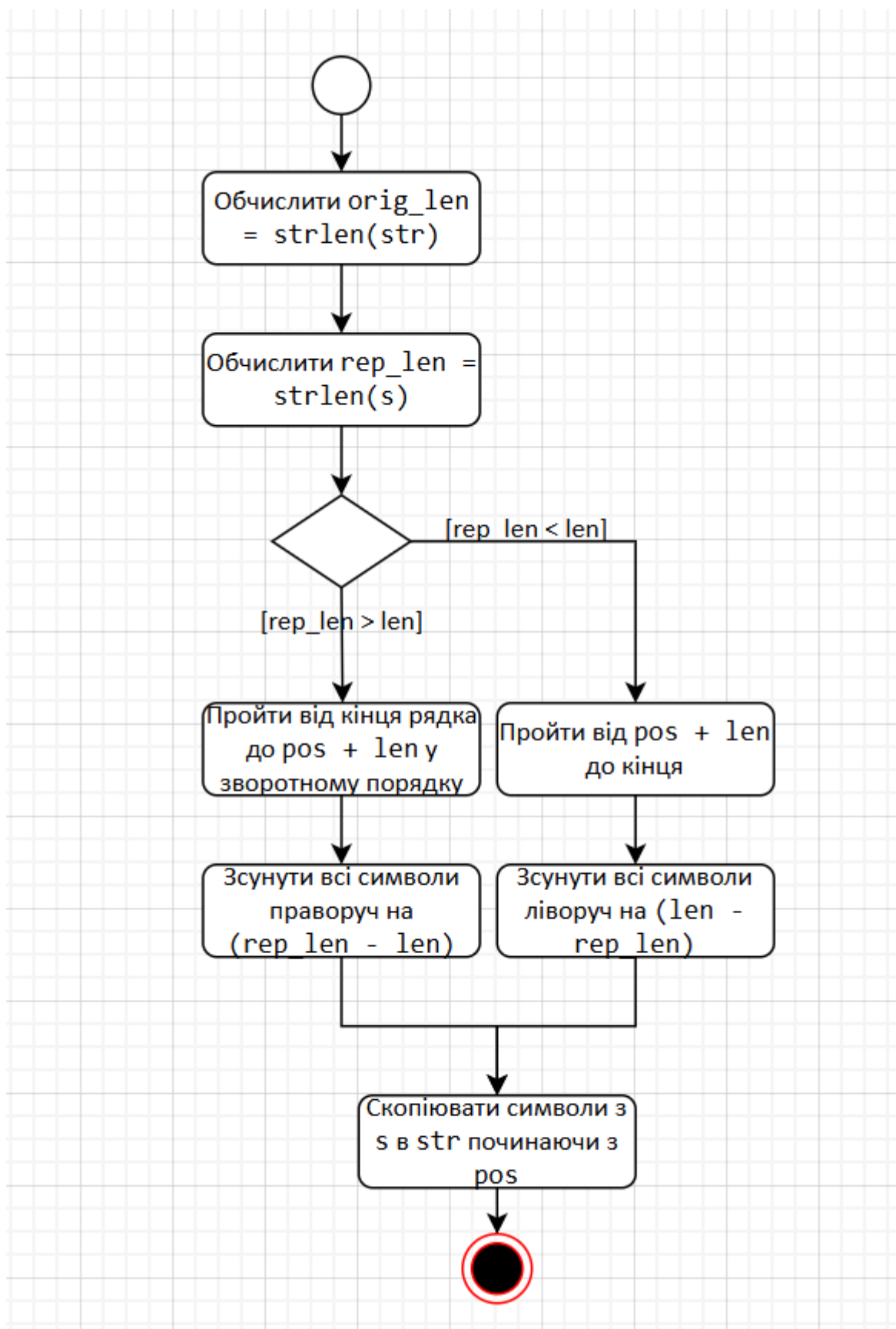


Рисунок 2 — заміна рядка без використання стандартних функцій .

На рис.3 показано діаграму коду програми для завдання 2.А,

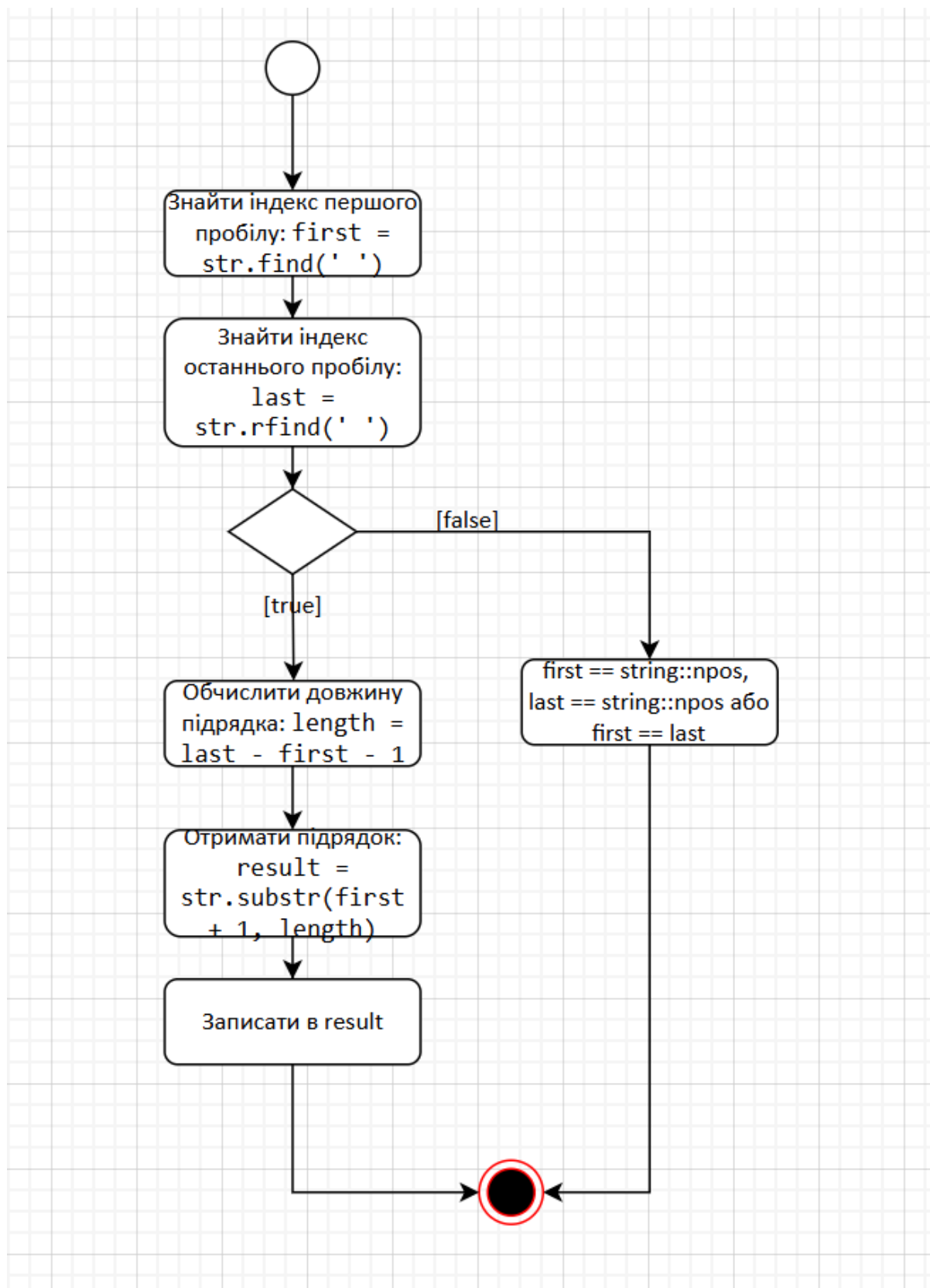


Рисунок 3 — повертає підрядок між першим і останнім пробілом у рядку.

На рис.4 показано діаграму коду програми для завдання 2.D

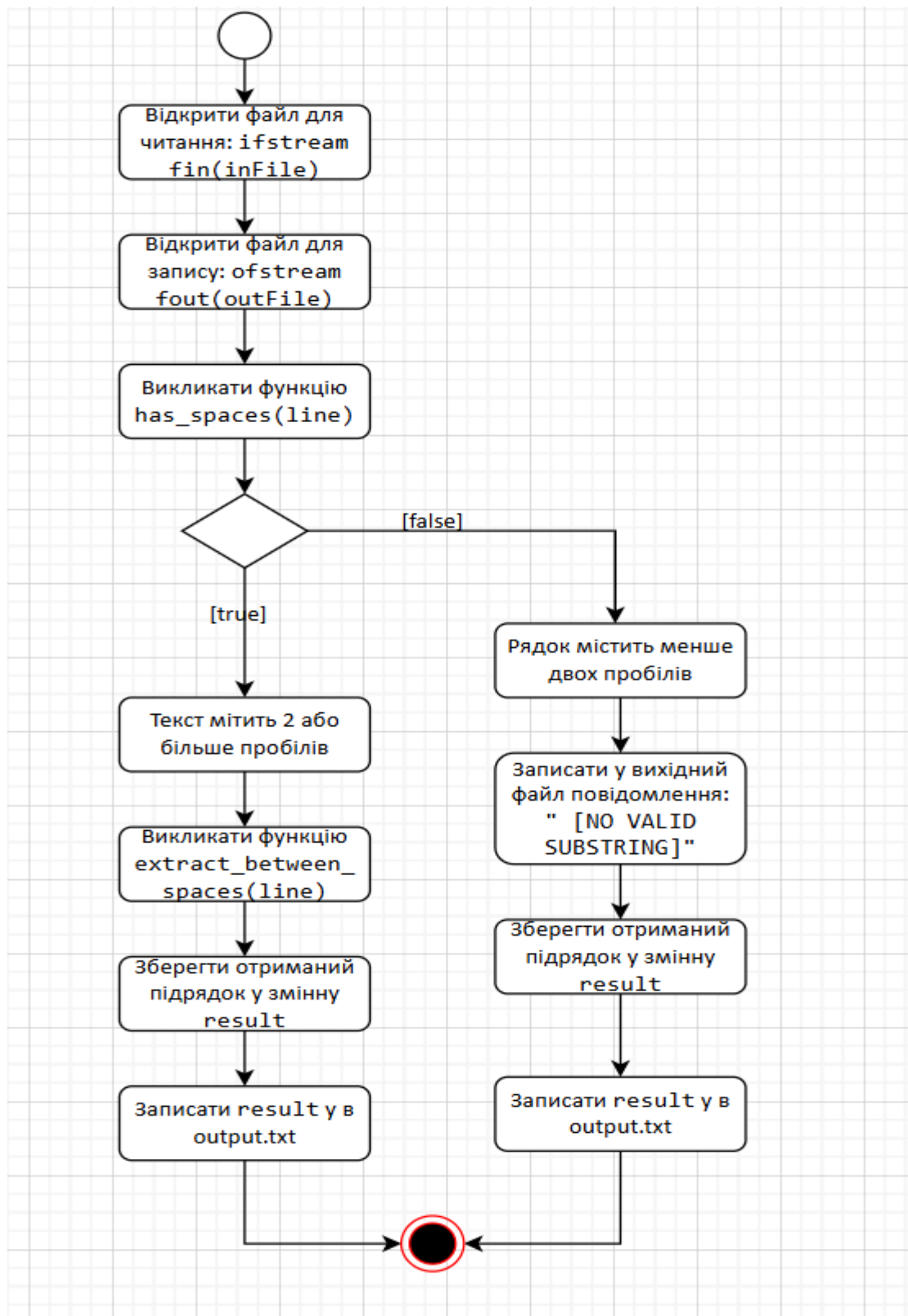


Рисунок 4 — обробляє файл рядок за рядком, виділяючи підрядки між пробілами й записуючи результат у новий файл.