

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.301.175.318.20 ЛР

Виконав студент гр. 318

Аліна ХОБОТ

(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.

Варіант 25. `size_t find (const char* s, size_t pos = 0) 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.

String48. Дано рядок, що складається з кирилических слів, набраних великими літерами і розділених пробілами (одним або декількома). Перетворити кожне слово в рядку, замінивши в ньому всі наступні входження його першої літери на символ «.» (крапка). Наприклад, слово «МИНИМУМ» треба перетворити в «МИНИ.У.». Кількість пробілів між словами не змінювати.

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.

п. 1.А

Опис функції:

Метод `find` шукає перше входження підрядка, заданого С-рядком (`const char* s`), у поточному об'єкті `string`, починаючи з позиції `pos`. Пошук є прямим - від початкової позиції до кінця рядка.

Опис параметрів:

`const char* s` - вказівник на С-рядок (масив символів, який закінчується нульовим символом `\0`), який потрібно знайти в рядку.

`size_t pos = 0` - (необов'язковий параметр) позиція в поточному рядку, з якої починається пошук. За замовчуванням - з початку рядка.

Опис результату:

Повертає індекс першого входження підрядка `s` у рядку, якщо такий знайдено.

Якщо входження не знайдено, повертається значення `string::npos` спеціальне значення, яке означає «не знайдено».

п. 1.В

Опис функції:

Функція `my_find` виконує пошук першого входження підрядка (`target`) у вихідному рядку (`source`), починаючи з вказаної позиції (`pos`).

Пошук виконується без використання функцій стандартної бібліотеки для обробки рядків.

Параметри:

`const char* source`

- Вказівник на вихідний С-рядок (масив символів), у якому виконується пошук.

`const char* target`

- Вказівник на шуканий підрядок (С-рядок), який необхідно знайти в `source`.

`size_t pos = 0`

- Початкова позиція в рядку `source`, з якої починається пошук (необов'язковий параметр, за замовчуванням 0).

Результат:

Якщо входження підрядка `target` знайдено, функція повертає індекс першого символу входження (тип `size_t`).

Якщо підрядок не знайдено, функція повертає `string::npos` — спеціальне значення, яке означає "не знайдено" (звичайно це -1, приведене до типу `size_t`).

Лістинг коду вирішення завдання таблиця 1, варіант 25 наведено в дод. А (стор. 7).

Екран роботи програми показаний на рис. Б.1 (дод. Б, стор. 17)

Завдання 2.

п. 2.А

Опис функції:

Функція `processString` приймає рядок, що складається з великих кириличних слів, розділених одним або кількома пробілами. Для кожного слова вона замінює всі повторні входження першої літери на крапку (.), зберігаючи кількість пробілів між словами. Повертає оброблений рядок як результат.

Параметри функції

`const string& input` – вхідний рядок, що містить одне або кілька слів, набраних великими літерами кирилицею, розділених одним або декількома пробілами. Рядок не містить початкових або кінцевих пробілів, але може містити кілька пробілів між словами.

Результат функції

`string` – новий рядок, у якому кожне слово оброблено наступним чином: усі входження першої літери (крім першого) замінено на крапку, при цьому кількість пробілів між словами збережено незмінною.

Лістинг коду вирішення завдання таблиця 2, варіант String48 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.2 (дод. Б, стор. 18)

п. 2.В

Назва функції:

`bool isValidInput(const string& input)`

Призначення:

Перевіряє, чи рядок складається лише з великих кириличних літер і пробілів, як вимагає умова задачі.

Параметри:

`input` — рядок для перевірки (тип `string`).

Результат:

true — якщо рядок коректний (слова з великих кириличних літер, розділені пробілами);

false — якщо рядок містить інші символи (малі літери, латиниця, знаки тощо).

Лістинг коду вирішення завдання п. 2.В наведено в дод. А (стор. 9).

п. 2.С

Лістинг файлу вирішення завдання п. 2.С показаний на рис. Б.3 (дод. Б, стор. 18)

п. 2.D

Лістинг коду вирішення завдання п. 2.D наведено в дод. А (стор. 9).

п. 2.E

Лістинг файлу вирішення завдання п. 2.E показаний на рис. Б.4 (дод. Б, стор. 18)

Завдання 3.

Лістинг коду вирішення завдання 3 наведено в дод. А (стор. 13).

Екран роботи програми показаний на рис. Б.5 (дод. Б, стор. 19)

ВИСНОВКИ

Було вивчено 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.

ДОДАТОК А

```

#include <iostream>
#include <Windows.h>

using namespace std;

size_t my_find(const char* source, const char* target, size_t pos = 0) {
    // Обчислення довжини source
    size_t source_len = 0;
    while (source[source_len] != '\0') source_len++;

    // Обчислення довжини target
    size_t target_len = 0;
    while (target[target_len] != '\0') target_len++;

    if (target_len == 0 || target_len > source_len) return string::npos;

    for (size_t i = pos; i <= source_len - target_len; ++i) {
        size_t j = 0;
        while (j < target_len && source[i + j] == target[j]) {
            ++j;
        }
        if (j == target_len) {
            return i; // Знайдено входження
        }
    }
    return string::npos; // Не знайдено
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(LC_ALL, "");

    const char* text = "Hello, world!";
    const char* word = "world";

    size_t result = my_find(text, word);

    if (result != string::npos) {
        cout << " Слово знайдено на позиції:: " << result << endl;
    } else {
        cout << "Не знайдено." << endl;
    }

    return 0;
}

```

Лістинг коду до завдання 2.А

```

#include <iostream>
#include <string>
#include <locale>
#include <Windows.h> // Додаємо для SetConsoleCP

using namespace std;

string processString(const string& input) {
    setlocale(LC_ALL, ""); // Встановлення локалі для роботи з кирилицею

    string result;
    size_t i = 0;

    while (i < input.length()) {
        // Копіюємо всі пробіли (щоб зберегти їх кількість між словами)
        while (i < input.length() && input[i] == ' ') {
            result += ' ';
            ++i;
        }

        // Початок слова
        size_t start = i;

        // Знаходимо кінець слова
        while (i < input.length() && input[i] != ' ') {
            ++i;
        }

        // Якщо слово знайдено
        if (start < i) {
            string word = input.substr(start, i - start);
            if (!word.empty()) {
                char firstChar = word[0];
                for (size_t j = 0; j < word.length(); ++j) {
                    if (j != 0 && word[j] == firstChar)
                        result += '.';
                    else
                        result += word[j];
                }
            }
        }
    }

    return result;
}

int main() {
    // Встановлення кодування консолі
    SetConsoleCP(1251);

```



```

SetConsoleOutputCP(1251);
setlocale(LC_ALL, "");

string input = "МИНИМУМ  ТЕСТ  ПРИКЛАД";
string output = processString(input);
cout << output << endl;
return 0;
}

```

Лістинг коду до завдання 2.В

```

// Функція перевірки вхідного рядка
bool isValidInput(const string& input) {
    for (char c : input) {
        if (c == ' ')
            continue;

        unsigned char uc = static_cast<unsigned char>(c);
        if (!(uc >= 192 && uc <= 223) || uc == 168 || uc == 170 || uc == 175 ||
uc == 178)) {
            return false;
        }
    }
    return true;
}

```

Лістинг коду до завдання 2.D

```

#include <iostream>
#include <string>
#include <locale>
#include <Windows.h>
#include <fstream>

using namespace std;

// Функція обробки рядка (заміна повторень першої літери на '.')
string processString(const string& input) {
    setlocale(LC_ALL, "");

    string result;
    size_t i = 0;

    while (i < input.length()) {
        // Копіюємо всі пробіли
        while (i < input.length() && input[i] == ' ') {
            result += ' ';
            ++i;
        }

        size_t start = i;

```

```

        while (i < input.length() && input[i] != ' ') {
            ++i;
        }

        if (start < i) {
            string word = input.substr(start, i - start);
            if (!word.empty()) {
                char firstChar = word[0];
                for (size_t j = 0; j < word.length(); ++j) {
                    if (j != 0 && word[j] == firstChar)
                        result += '.';
                    else
                        result += word[j];
                }
            }
        }
    }

    return result;
}

// Функція перевірки вхідного рядка
bool isValidInput(const string& input) {
    for (char c : input) {
        if (c == ' ')
            continue;

        unsigned char uc = static_cast<unsigned char>(c);
        if (!(uc >= 192 && uc <= 223) || uc == 168 || uc == 170 || uc == 175 ||
uc == 178)) {
            return false;
        }
    }
    return true;
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(LC_ALL, "");

    // Створення текстового файлу з 10 прикладами
    ofstream outFile("input.txt");
    if (outFile.is_open()) {
        outFile << "МИНИМУМ ТЕСТ ПРИКЛАД\n";
        outFile << "AAAA BBB BBBB\n";
        outFile << "СЛОВО  ЗРАЗОК\n";
        outFile << "ПРИВІТ  СВІТ\n";
        outFile << "ТЕСТУВАННЯ КОДУ\n";
        outFile << "КОНТРОЛЬНА  РОБОТА\n";
    }
}

```

```

    outFile << "ДАНІ  ІНФОРМАЦІЯ\n";
    outFile << "КОМПІЛЯТОР  ПРОГРАМА\n";
    outFile << "КИРИЛИЦЯ  ПРОБІЛИ\n";
    outFile << "СТРОКА  ПЕРЕВІРКА\n";
    outFile.close();
    cout << "Файл 'input.txt' створено з прикладами рядків.\n";
}
else {
    cout << "Помилка відкриття файлу для запису.\n";
}

int choice;
string input;

do {
    cout << "\n=== МЕНЮ ===\n";
    cout << "1. Ввести рядок\n";
    cout << "2. Перевірити рядок\n";
    cout << "3. Обробити рядок\n";
    cout << "4. Обробити файл та зберегти результат\n";
    cout << "0. Вихід\n";
    cout << "Ваш вибір: ";
    cin >> choice;
    cin.ignore(); // для очищення буфера після вводу числа

    switch (choice) {
    case 1:
        cout << "Введіть рядок: ";
        getline(cin, input);
        break;

    case 2:
        if (input.empty()) {
            cout << "Спочатку введіть рядок.\n";
        }
        else if (isValidInput(input)) {
            cout << "Рядок коректний.\n";
        }
        else {
            cout << "Рядок містить некоректні символи.\n";
        }
        break;

    case 3:
        if (input.empty()) {
            cout << "Спочатку введіть рядок.\n";
        }
        else if (!isValidInput(input)) {
            cout << "Рядок некоректний. Обробка неможлива.\n";
        }
    }
}

```

```

else {
    string output = processString(input);
    cout << "Результат: " << output << endl;
}
break;
case 4: {
    string inputFileName, outputFileName;
    cout << "Введіть назву вхідного файлу: ";
    getline(cin, inputFileName);
    cout << "Введіть назву вихідного файлу: ";
    getline(cin, outputFileName);

    ifstream inFile(inputFileName);
    ofstream outFile(outputFileName);

    if (!inFile.is_open()) {
        cout << "Не вдалося відкрити вхідний файл.\n";
        break;
    }

    if (!outFile.is_open()) {
        cout << "Не вдалося відкрити вихідний файл.\n";
        break;
    }

    string line;
    while (getline(inFile, line)) {
        if (isValidInput(line)) {
            outFile << processString(line) << endl;
        }
        else {
            outFile << "[Некоректний рядок]" << endl;
        }
    }

    cout << "Обробка завершена. Результат записано у файл '" <<
outputFileName << "'.\n";

    inFile.close();
    outFile.close();
    break;
}

case 0:
    cout << "Завершення програми.\n";
    break;

default:
    cout << "Невірний вибір. Спробуйте ще раз.\n";
}

```

```

    } while (choice != 0);

    return 0;
}

```

Лістинг коду до завдання 3

task1.h:

```
#pragma once
```

```
void runTask1(); // Реалізація пошуку підрядка
```

task1.cpp:

```
#include <iostream>
```

```
#include <Windows.h>
```

```
#include <string>
```

```
using namespace std;
```

```

void runTask1() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(LC_ALL, "");

    const char* text = "Hello, world!";
    const char* word = "world";

    size_t source_len = 0;
    while (text[source_len] != '\0') source_len++;

    size_t target_len = 0;
    while (word[target_len] != '\0') target_len++;

    size_t result;
    if (target_len == 0 || target_len > source_len) {
        result = string::npos;
    }
    else {
        result = string::npos;
        for (size_t i = 0; i <= source_len - target_len; ++i) {
            size_t j = 0;
            while (j < target_len && text[i + j] == word[j]) {
                ++j;
            }
            if (j == target_len) {
                result = i;
                break;
            }
        }
    }
}

```

```

    if (result != string::npos) {
        cout << "Слово знайдено на позиції: " << result << endl;
    }
    else {
        cout << "Не знайдено." << endl;
    }
}

```

task2.h:

```

#pragma once

```

```

void runTask2(); // Меню для обробки рядків

```

task2.cpp:

```

#include <iostream>
#include <string>
#include <fstream>
#include <Windows.h>
using namespace std;

```

```

void runTask2() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(LC_ALL, "");

    string input;
    int choice;

    do {
        cout << "\n--- Меню Завдання 2 ---\n";
        cout << "1. Ввести рядок\n";
        cout << "2. Перевірити рядок\n";
        cout << "3. Обробити рядок\n";
        cout << "4. Обробити файл\n";
        cout << "0. Назад до головного меню\n";
        cout << "Ваш вибір: ";
        cin >> choice;
        cin.ignore();

        switch (choice) {
            case 1:
                cout << "Введіть рядок: ";
                getline(cin, input);
                break;

            case 2: {
                if (input.empty()) {
                    cout << "Спочатку введіть рядок.\n";
                    break;
                }
            }

```

```

        bool valid = true;
        for (char c : input) {
            if (c == ' ') continue;
            unsigned char uc = static_cast<unsigned char>(c);
            if (!(uc >= 192 && uc <= 223) || uc == 168 || uc == 170 || uc ==
175 || uc == 178)) {
                valid = false;
                break;
            }
        }
        cout << (valid ? "Рядок коректний.\n" : "Некоректні символи у
рядку.\n");
        break;
    }

    case 3: {
        if (input.empty()) {
            cout << "Спочатку введіть рядок.\n";
            break;
        }

        bool valid = true;
        for (char c : input) {
            if (c == ' ') continue;
            unsigned char uc = static_cast<unsigned char>(c);
            if (!(uc >= 192 && uc <= 223) || uc == 168 || uc == 170 || uc ==
175 || uc == 178)) {
                valid = false;
                break;
            }
        }

        if (!valid) {
            cout << "Некоректний рядок.\n";
            break;
        }

        string result;
        size_t i = 0;
        while (i < input.length()) {
            while (i < input.length() && input[i] == ' ') {
                result += ' ';
                ++i;
            }

            size_t start = i;
            while (i < input.length() && input[i] != ' ') {
                ++i;
            }

```

```

        if (start < i) {
            string word = input.substr(start, i - start);
            if (!word.empty()) {
                char firstChar = word[0];
                for (size_t j = 0; j < word.length(); ++j) {
                    if (j != 0 && word[j] == firstChar)
                        result += '.';
                    else
                        result += word[j];
                }
            }
        }
    }
    cout << "Результат: " << result << endl;
    break;
}

case 4: {
    string inName, outName;
    cout << "Ім'я вхідного файлу: "; getline(cin, inName);
    cout << "Ім'я вихідного файлу: "; getline(cin, outName);

    ifstream inFile(inName);
    ofstream outFile(outName);

    if (!inFile || !outFile) {
        cout << "Помилка відкриття файлів.\n";
        break;
    }
    string line;
    while (getline(inFile, line)) {
        bool valid = true;
        for (char c : line) {
            if (c == ' ') continue;
            unsigned char uc = static_cast<unsigned char>(c);
            if (!(uc >= 192 && uc <= 223) || uc == 168 || uc == 170 ||
uc == 175 || uc == 178)) {
                valid = false;
                break;
            }
        }
        if (!valid) {
            outFile << "[Некоректний рядок]" << endl;
        }
        else {
            string result;
            size_t i = 0;
            while (i < line.length()) {
                while (i < line.length() && line[i] == ' ') {
                    result += ' ';

```



```

        ++i;
    }
    size_t start = i;
    while (i < line.length() && line[i] != ' ') {
        ++i;
    }
    if (start < i) {
        string word = line.substr(start, i - start);
        if (!word.empty()) {
            char firstChar = word[0];
            for (size_t j = 0; j < word.length(); ++j) {
                if (j != 0 && word[j] == firstChar)
                    result += '.';
                else
                    result += word[j];
            }
        }
    }
    outFile << result << endl;
}

}
cout << "Файл оброблено.\n";
break;
}
case 0:
    cout << "Повернення до головного меню.\n";
    break;
default:
    cout << "Невірний вибір.\n";
}

} while (choice != 0);
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програм



Рисунок Б.1 – Екран виконання програми для вирішення завдання таблиця 1, варіант 25.



Рисунок Б.2 – Екран виконання програми для вирішення завдання таблиця 2, варіант String48.

```
input.txt – Блокнот
Файл  Правка  Формат  Вид
МИНИМУМ ТЕСТ ПРИКЛАД
АААА БББ ВВВВ
СЛОВО ЗРАЗОК
ПРИВІТ СВІТ
ТЕСТУВАННЯ КОДУ
КОНТРОЛЬНА РОБОТА
ДАНІ ІНФОРМАЦІЯ
КОМПІЛЯТОР ПРОГРАМА
КИРИЛИЦЯ ПРОБІЛИ
СТРОКА ПЕРЕВІРКА
```

Рисунок Б.3 – Лістинг файлу вирішення завдання п. 2.С

```
output.txt – Блокнот
Файл  Правка  Формат  Вид
МИНИ.У. ТЕС. ПРИКЛАД
А... Б.. В...
СЛОВО ЗРА.ОК
ПРИВІТ СВІТ
ТЕС.УВАННЯ КОДУ
КОНТРОЛЬНА РОБОТА
ДАНІ ІНФОРМАЦ.Я
КОМПІЛЯТОР ПРОГРАМА
КИРИЛИЦЯ ПРОБІЛИ
СТРОКА ПЕРЕВІРКА
```

Рисунок Б.4 – Лістинг файлу вирішення завдання п. 2.Е

Консоль отладки Microsoft Visual Studio

```

=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
0. Вихід
Ваш вибір: 1
Слово знайдено на позиції: 7

=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
0. Вихід
Ваш вибір: 2

--- Меню Завдання 2 ---
1. Ввести рядок
2. Перевірити рядок
3. Обробити рядок
4. Обробити файл
0. Назад до головного меню
Ваш вибір: 1
Введіть рядок: ДИДА

--- Меню Завдання 2 ---
1. Ввести рядок
2. Перевірити рядок
3. Обробити рядок
4. Обробити файл
0. Назад до головного меню
Ваш вибір: 3
Результат: ДИ.А

--- Меню Завдання 2 ---
1. Ввести рядок
2. Перевірити рядок
3. Обробити рядок
4. Обробити файл
0. Назад до головного меню
Ваш вибір: 4
Ім'я вхідного файлу: input.txt
Ім'я вихідного файлу: output.txt
Файл оброблено.

--- Меню Завдання 2 ---
1. Ввести рядок
2. Перевірити рядок
3. Обробити рядок
4. Обробити файл
0. Назад до головного меню
Ваш вибір: 0
Повернення до головного меню.

=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
0. Вихід
Ваш вибір: 0
Завершення програми.

```

input.txt – Блокнот

```

Файл  Правка  Формат  Вид  Справка
МИНИМУМ ТЕСТ ПРИКЛАД
АААА БББ ВВВВ
СЛОВО ЗРАЗОК
ПРИВІТ СВІТ
ТЕСТУВАННЯ КОДУ
КОНТРОЛЬНА РОБОТА
ДАНІ ІНФОРМАЦІЯ
КОМПІЛЯТОР ПРОГРАМА
КИРИЛИЦЯ ПРОБІЛИ
СТРОКА ПЕРЕВІРКА

```

output.txt – Блокнот

```

Файл  Правка  Формат  Вид  Справка
МИНИ.У. ТЕС. ПРИКЛАД
А... Б.. В...
СЛОВО ЗРА.ОК
ПРИВІТ СВІТ
ТЕС.УВАННЯ КОДУ
КОНТРОЛЬНА РОБОТА
ДАНІ ІНФОРМАЦ.Я
КОМПІЛЯТОР ПРОГРАМА
КИРИЛИЦЯ ПРОБІЛИ
СТРОКА ПЕРЕВІРКА

```

Рисунок Б.5 – Екран виконання програми для вирішення завдання 3

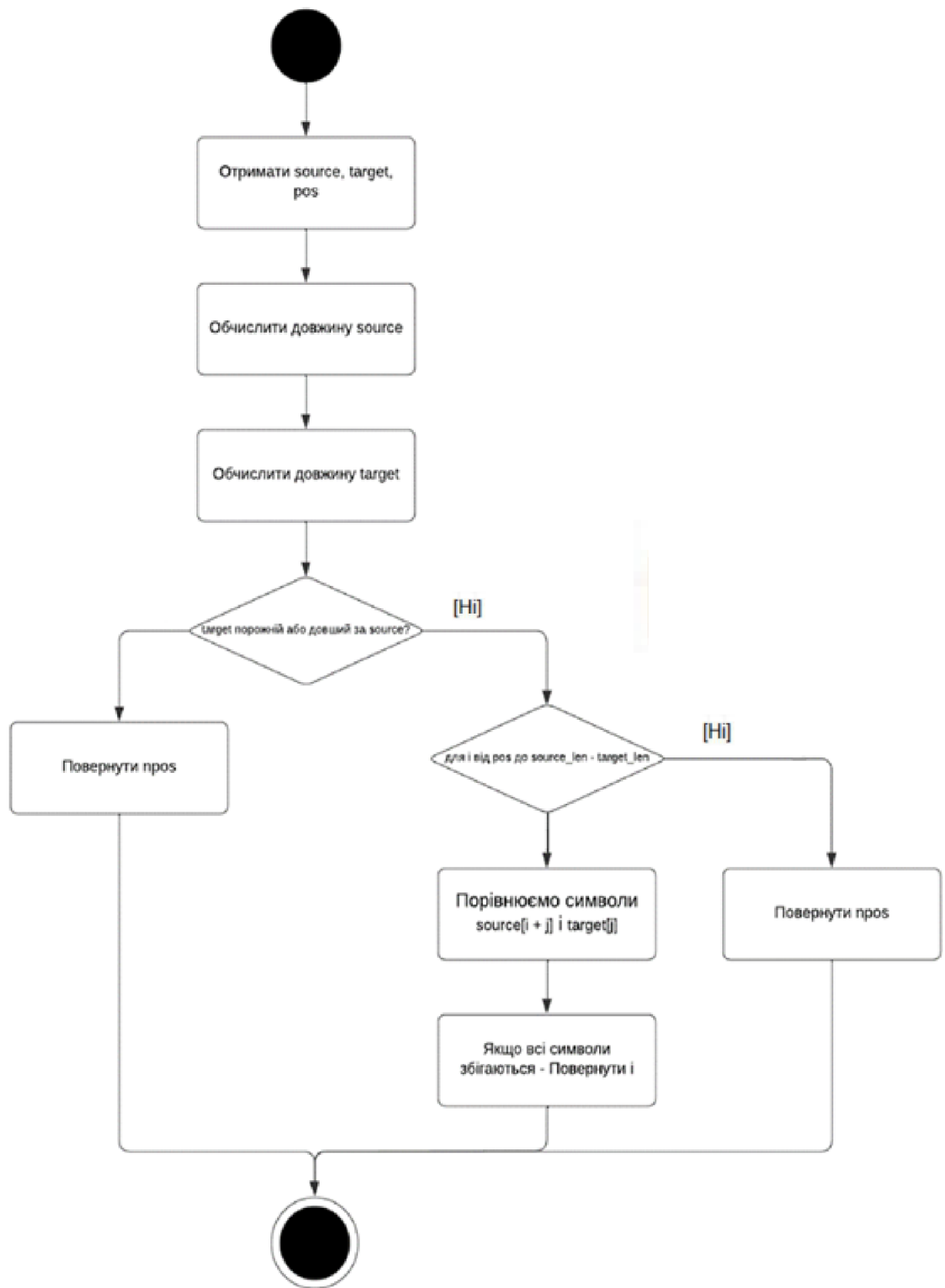


Рисунок Б.6 – Діаграма для завдання таблиця 1, варіант 25 п.1.В

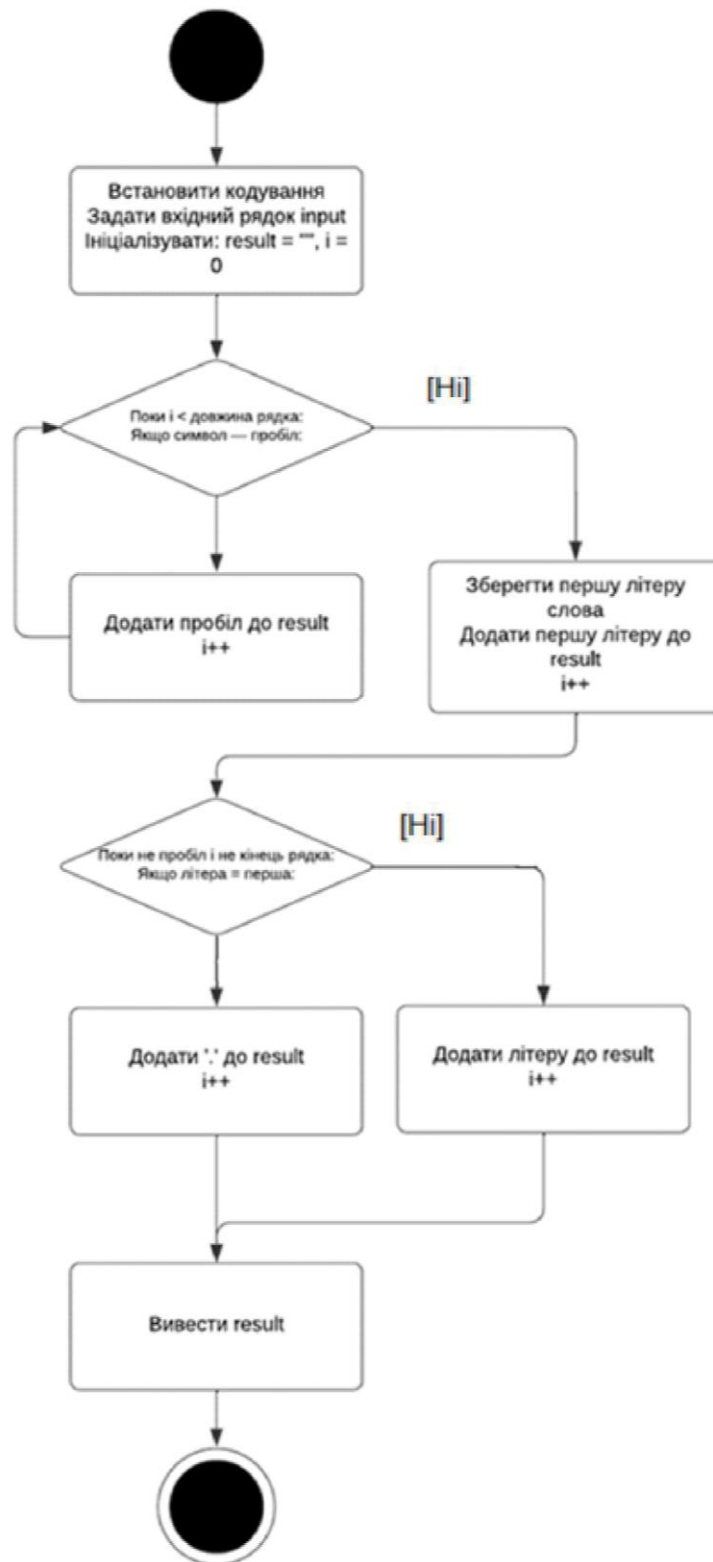


Рисунок Б.7 – Діаграма для завдання таблиця 2, варіант String48 п. 2.А

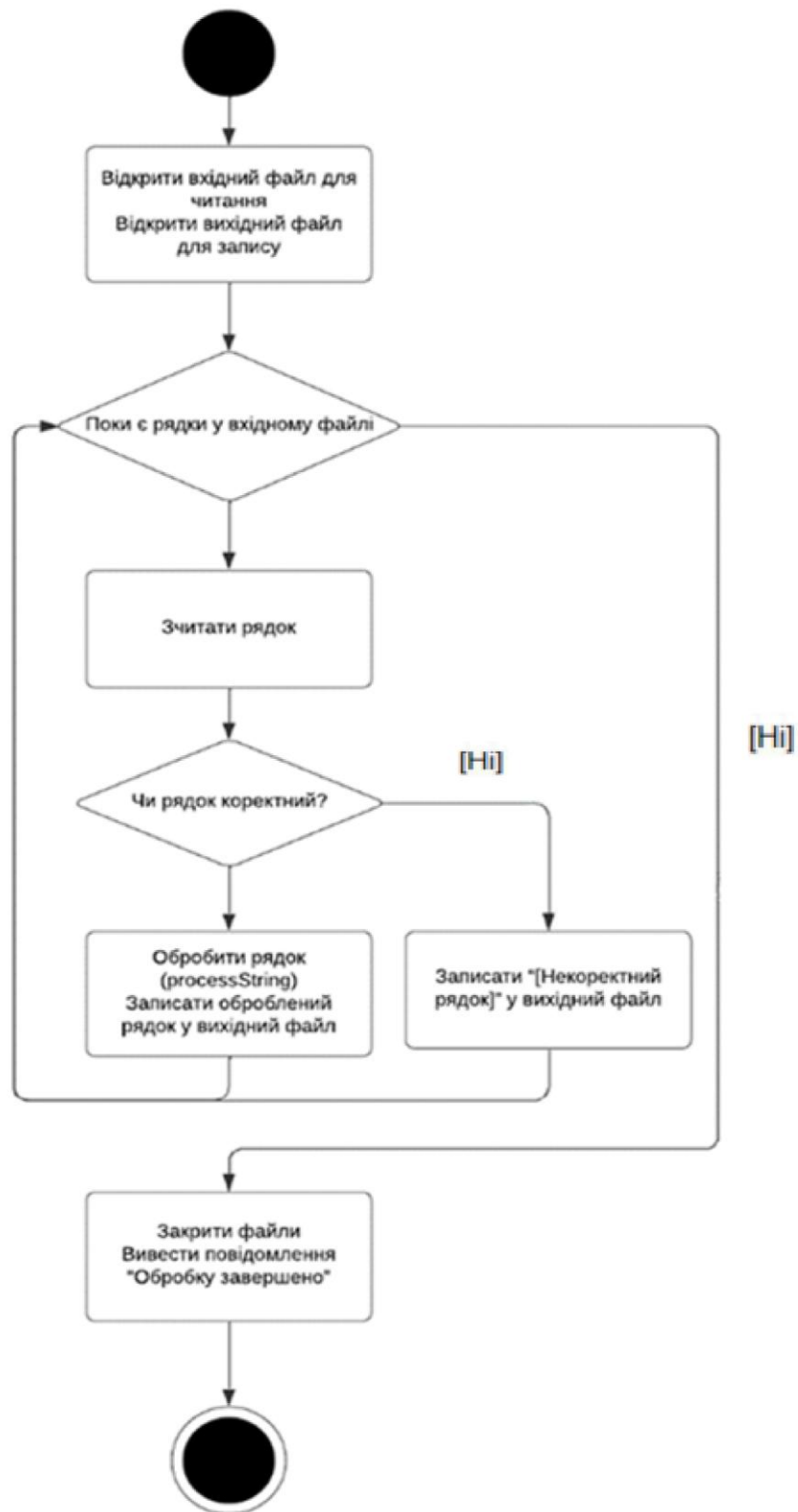


Рисунок Б.8 – Діаграма для завдання таблиця 2, варіант String48 п. 2.D