

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.301.312.2 ЛР

Виконав студент гр. 312

\_\_\_\_\_ Ксенія ВЕЛІКОДАНОВА \_\_\_\_\_  
(підпис, дата) (П.І.Б.)

Перевірів

\_\_\_\_\_ к.т.н., доц. Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

2023

## 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.

- A. Вивчити по документації метод стандартного класу `string` `size_t find_last_of (const string& str, size_t pos = npos) const`;
- 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`. 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 String61 з табл.2. 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.A і 2.B, обробити текстовий. Додаткові дані ввести з консолі.
- E. 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.

### Варіант 2

- 1 Завдання 36, табл. 1
- 2 Завдання String61, табл. 2

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі 36

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

`const char* src,`

`const char* chars,`

`size_t pos`

Вихідні дані (ім'я, опис, тип):

`size_t`

Алгоритм вирішення показано на рис. 1

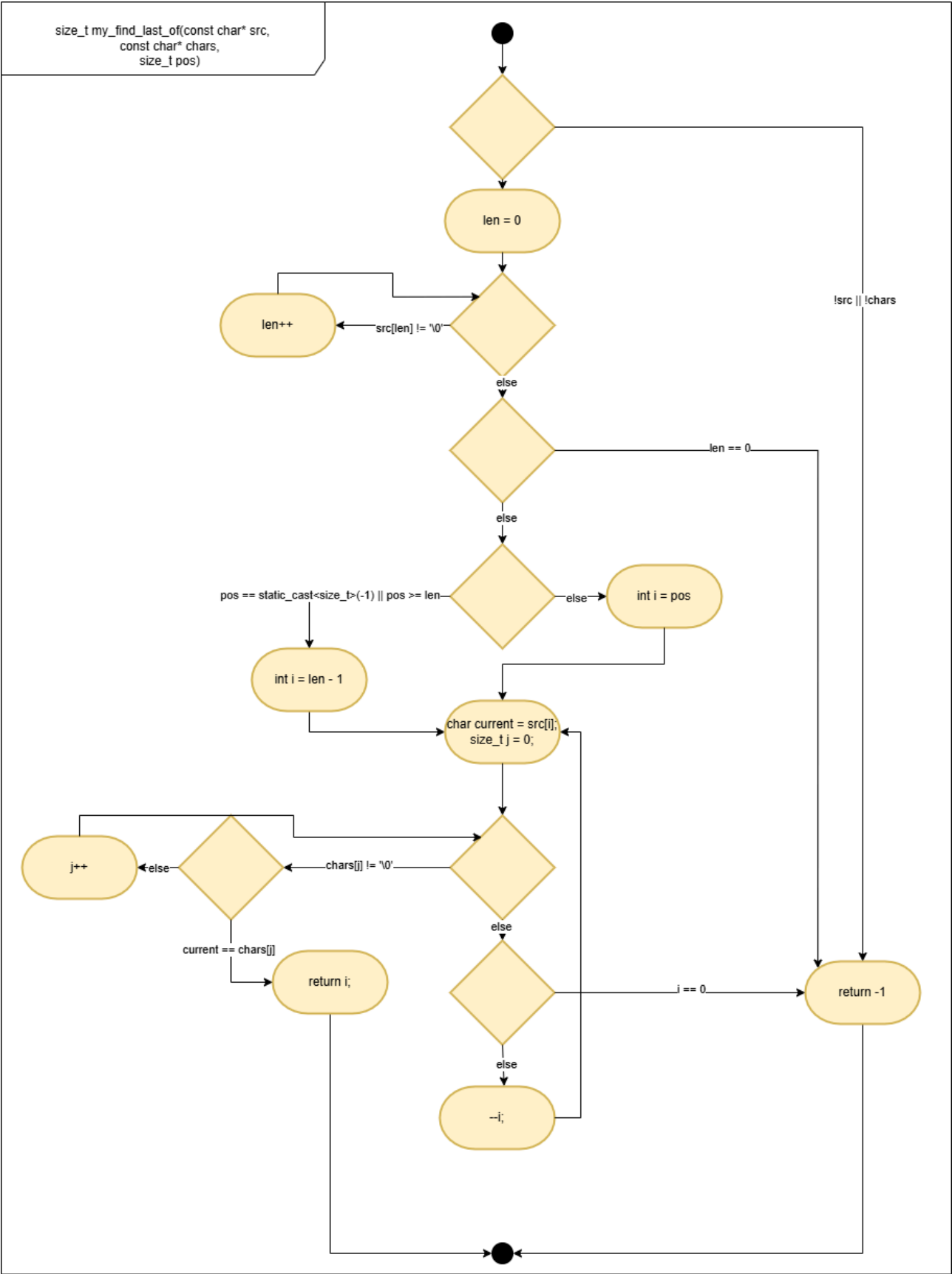


Рисунок 1 – Алгоритм рішення 36 задачі

Лістинг коду вирішення задачі 36 наведено в дод. А (стор. 7).  
Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі String61.

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

`const std::string& path`

Вихідні дані (ім'я, опис, тип):

`std::string`

Алгоритм вирішення показано на рис. 2

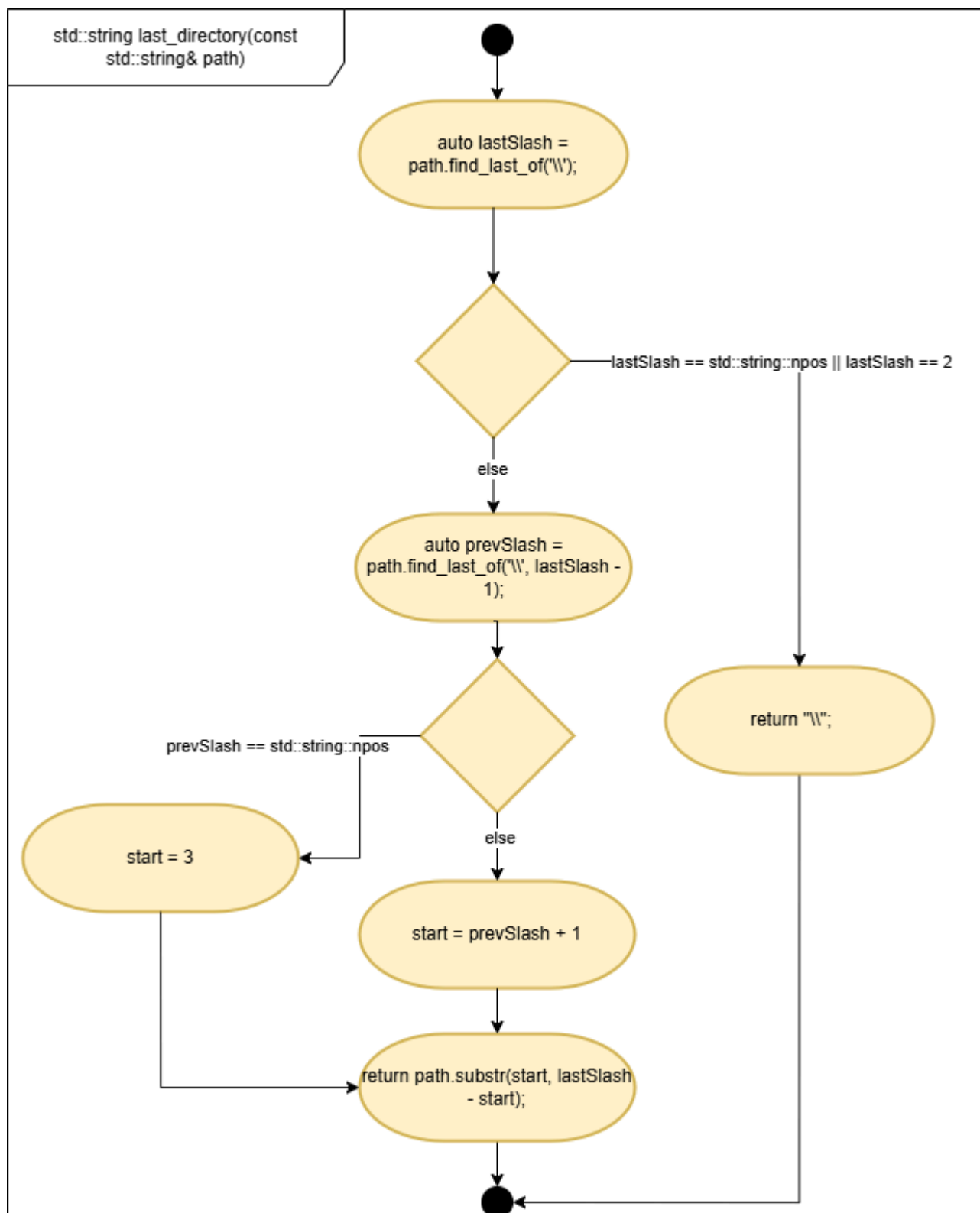


Рисунок 2 – Алгоритм String61

Лістинг коду вирішення задачі String61 наведено в дод. А (стор. 7).  
Екран роботи програми показаний на рис. Б.2.

## ВИСНОВКИ

Було вивчено принципи роботи з рядками без використання стандартних функцій. Закріплено на практиці навички обробки текстових файлів і реалізації пошуку в масиві символів. Виникли труднощі з перевіркою коректності шляхів, які були вирішені шляхом ручної валідації структури рядка.

## ДОДАТОК А

### Лістинг коду програми

```

main.cpp
#include <iostream>
#include "FindLastOf.h"
#include "PathUtilities.h"
#include <iostream>
#include <fstream>
#include <string>

using namespace std;
// — Task 1 —————
void run_task1()
{
    cout << "\n=== Task 1: find_last_of demo ===\n"
          << "src string : ";
    string src;
    getline(cin, src);

    cout << "chars to find : ";
    string chars; getline(cin, chars);

    size_t pos_my = my_find_last_of(src.c_str(), chars.c_str());
    size_t pos_std = src.find_last_of(chars);

    cout << "my_find_last_of -> " << pos_my << '\n'
          << "std::string ver. -> " << pos_std << '\n';
}

// — Task 2 —————
void run_task2()
{
    cout << "\n=== Task 2: process file paths ===\n"
          << "Input file (default \"input.txt\"): ";
    string inName; getline(cin, inName);
    if (inName.empty()) inName = "input.txt";

    ifstream fin(inName);
    if (!fin) { cerr << "Cannot open " << inName << '\n'; return; }

    ofstream fout("output.txt");

    string line;
    while (getline(fin, line))
    {
        if (is_valid_path(line))

```



```

        fout << last_directory(line) << '\n';
    else
        fout << "[invalid]\n";
    }
    cout << "Done. See output.txt\n";
}

int main()
{
    int choice = -1;

    do
    {
        cout << "\n===== MENU =====\n"
              "1. Task 1 - my_find_last_of demo\n"
              "2. Task 2 - file-path processing\n"
              "0. Exit\n"
              ">>> ";
        if (!(cin >> choice)) break;
        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // flush

        switch (choice)
        {
            case 1: run_task1(); break;
            case 2: run_task2(); break;
            case 0: break;
            default: cout << "Unknown option\n";
        }
    } while (choice != 0);

    return 0;
}

PathUtilities.h
#pragma once
#include <string>

/// 2-В. Перевірка, чи є 'path' повним шляхом виду «X:\dir\file.ext».
bool is_valid_path(const std::string& path);

/// 2-А. Повертає назву останнього каталогу або "\" для кореня.
std::string last_directory(const std::string& path);
PathUtilities.cpp
#include "PathUtilities.h"

// — допоміжна —
static bool is_alpha(char c)
{
    return (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z');
}

```

```

}

// 2-B
bool is_valid_path(const std::string& path)
{
    if (path.size() < 4) return false;           // "C:\a"
    if (!is_alpha(path[0]) || path[1] != ':' || path[2] != '\\')
        return false;

    auto lastSlash = path.find_last_of('\\');
    auto lastDot = path.find_last_of('.');

    // файл + розширення після останнього '\'
    return lastDot != std::string::npos &&
        lastDot > lastSlash + 1;
}

// 2-A
std::string last_directory(const std::string& path)
{
    auto lastSlash = path.find_last_of('\\');
    if (lastSlash == std::string::npos || lastSlash == 2)
        return "\\";           // корінь: "C:\file.ext"

    auto prevSlash = path.find_last_of('\\', lastSlash - 1);
    std::size_t start = (prevSlash == std::string::npos ? 3 : prevSlash + 1);
    return path.substr(start, lastSlash - start);
}

FindLastOf.h
#pragma once
#include <cstdint>    // size_t

/// Повертає індекс останнього символу з 'chars' у рядку 'src',
/// починаючи з позиції 'pos' і рухаючись ліворуч.
/// Якщо не знайдено – повертає size_t(-1).
size_t my_find_last_of(const char* src,
    const char* chars,
    size_t pos = static_cast<size_t>(-1));

FindLastOf.cpp
#include "FindLastOf.h"

size_t my_find_last_of(const char* src,
    const char* chars,
    size_t pos)
{
    if (!src || !chars) return static_cast<size_t>(-1);

    // — довжина src —————
    size_t len = 0;
    while (src[len] != '\\0') ++len;

```

```

if (len == 0) return static_cast<size_t>(-1);

size_t i = (pos == static_cast<size_t>(-1) || pos >= len) ? len - 1 : pos;

// — пошук з кінця —————
for (;;)
{
    char current = src[i];

    // чи входить current у chars?
    for (size_t j = 0; chars[j] != '\0'; ++j)
        if (current == chars[j])
            return i;

    if (i == 0) break;    // вийти, щойно дійшли до початку
    --i;
}
return static_cast<size_t>(-1);
}

```

## ДОДАТОК Б

### Скрін-шоти вікна виконання програми

```

===== MENU =====
1. Task 1  û my_find_last_of demo
2. Task 2  û file-path processing
0. Exit
>>> 1

=== Task 1: find_last_of demo ===
src string : Hello World
chars to find : l
my_find_last_of  -> 9
std::string ver. -> 9

```

Рисунок Б.1 – Екран виконання програми для вирішення завдання

36

<pre> ===== MENU ===== 1. Task 1  û my_find_last_of demo 2. Task 2  û file-path processing 0. Exit &gt;&gt;&gt; 2  === Task 2: process file paths === Input file (default "input.txt"): Done. See output.txt </pre>	<table border="1"> <thead> <tr> <th>output.txt</th> <th>input.txt</th> <th>FindLastOf.cpp</th> </tr> </thead> <tbody> <tr><td>1</td><td>C:\Alt\Docs\report.docx</td><td></td></tr> <tr><td>2</td><td>C:\Games\Game.exe</td><td></td></tr> <tr><td>3</td><td>D:\Music\Rock\song.mp3</td><td></td></tr> <tr><td>4</td><td>E:\file_only.txt</td><td></td></tr> <tr><td>5</td><td>X:\One\Two\Three\data.bin</td><td></td></tr> <tr><td>6</td><td>Z:\.\broken\path</td><td></td></tr> <tr><td>7</td><td>C:\file_without_drive_slash.txt</td><td></td></tr> <tr><td>8</td><td>F:\RootOnly\photo.jpg</td><td></td></tr> <tr><td>9</td><td>C:\demo\last\spring\image.png</td><td></td></tr> <tr><td>10</td><td>C:\bad\no_extension</td><td></td></tr> </tbody> </table>	output.txt	input.txt	FindLastOf.cpp	1	C:\Alt\Docs\report.docx		2	C:\Games\Game.exe		3	D:\Music\Rock\song.mp3		4	E:\file_only.txt		5	X:\One\Two\Three\data.bin		6	Z:\.\broken\path		7	C:\file_without_drive_slash.txt		8	F:\RootOnly\photo.jpg		9	C:\demo\last\spring\image.png		10	C:\bad\no_extension		<table border="1"> <thead> <tr> <th>output.txt</th> <th>input.txt</th> <th>FindLastOf.cpp</th> </tr> </thead> <tbody> <tr><td>1</td><td>Docs</td><td></td></tr> <tr><td>2</td><td>Games</td><td></td></tr> <tr><td>3</td><td>Rock</td><td></td></tr> <tr><td>4</td><td>\</td><td></td></tr> <tr><td>5</td><td>Three</td><td></td></tr> <tr><td>6</td><td>[invalid]</td><td></td></tr> <tr><td>7</td><td>[invalid]</td><td></td></tr> <tr><td>8</td><td>RootOnly</td><td></td></tr> <tr><td>9</td><td>spring</td><td></td></tr> <tr><td>10</td><td>[invalid]</td><td></td></tr> </tbody> </table>	output.txt	input.txt	FindLastOf.cpp	1	Docs		2	Games		3	Rock		4	\		5	Three		6	[invalid]		7	[invalid]		8	RootOnly		9	spring		10	[invalid]	
output.txt	input.txt	FindLastOf.cpp																																																																		
1	C:\Alt\Docs\report.docx																																																																			
2	C:\Games\Game.exe																																																																			
3	D:\Music\Rock\song.mp3																																																																			
4	E:\file_only.txt																																																																			
5	X:\One\Two\Three\data.bin																																																																			
6	Z:\.\broken\path																																																																			
7	C:\file_without_drive_slash.txt																																																																			
8	F:\RootOnly\photo.jpg																																																																			
9	C:\demo\last\spring\image.png																																																																			
10	C:\bad\no_extension																																																																			
output.txt	input.txt	FindLastOf.cpp																																																																		
1	Docs																																																																			
2	Games																																																																			
3	Rock																																																																			
4	\																																																																			
5	Three																																																																			
6	[invalid]																																																																			
7	[invalid]																																																																			
8	RootOnly																																																																			
9	spring																																																																			
10	[invalid]																																																																			

Рисунок Б.2 – Екран виконання програми для вирішення завдання  
String 61