

Project: Transmission File and Malicious Code Analysis

Student Name: Ricardo Alfredo Calvo Pérez **Student ID:** A01028889 **Date:** 10/28/2024

Problem Description

This project aims to develop a C++ program that reads five fixed text files containing data on transmissions and malicious code. The program will determine if the malicious code is contained within the transmission data. Additionally, it will identify mirrored patterns (palindromes) and calculate the longest common substring between two transmissions.

Involved Files

1. Transmission Files:

- `transmission1.txt`
- `transmission2.txt`

2. Malicious Code Files:

- `mcode1.txt`
- `mcode2.txt`
- `mcode3.txt`

File Descriptions

- The **transmissionX.txt** files contain character sequences representing transmission data between devices.
- The **mcodeX.txt** files contain sequences of malicious code that need to be searched within the transmission files.

Program Objectives

Part 1: Malicious Code Detection

The program should analyze whether the contents of `mcode1.txt`, `mcode2.txt`, and `mcode3.txt` are found within `transmission1.txt` and `transmission2.txt`. The result should indicate `true` or `false` depending on whether the character sequence from each `mcodeX.txt` file is contained in the transmission, along with the starting position.

Output Format:

```
(true | false) starting_position
```

Example:

For `mcode1.txt` containing `A12B34` and `transmission1.txt` containing `ABCDEFGGFEDCBA12345`:

```
true 4  
false
```

Part 2: Palindrome Detection

The program should search for mirrored patterns (palindromes) within the transmission files. For each transmission file, it should display the starting and ending positions (starting from 1) of the longest palindrome found.

Output Format:

```
starting_position ending_position
```

Example:

For `transmission1.txt` containing `ABCDEFGGFEDCBA12345`:

```
1 13
```

Part 3: Longest Common Substring Detection

The program should compare the transmission files `transmission1.txt` and `transmission2.txt` to find the longest common substring. It should display the starting and ending positions (starting from 1) of the longest common substring between the two transmission files.

Output Format:

```
starting_position ending_position
```

Example:

For `transmission1.txt` containing `ABCDEFGGFEDCBA12345` and `transmission2.txt` containing `DEF123456789ABC123321DEF`:

```
14 18
```

Input

- There is no user input. The required files should be located in the same directory as the program.

Output

Part 1:

Displays whether each transmission file contains the codes from `mcode1.txt`, `mcode2.txt`, and `mcode3.txt`, along with the starting position for each code within the transmission.

Part 2:

Displays the starting and ending positions of the longest palindrome within each transmission file.

Part 3:

Displays the starting and ending positions of the longest common substring between the two transmission files.

Examples

Example 1 - Part 1

Files:

- `mcode1.txt`: A12B34
- `transmission1.txt`: ABCDEFGFEDCBA12345

Output:

```
true 13
```

Example 2 - Part 2

File:

- `transmission1.txt`: ABCDEFGFEDCBA12345

Output:

```
1 13
```

Example 3 - Part 3

Files:

- `transmission1.txt`: ABCDEFGFEDCBA12345
- `transmission2.txt`: DEF123456789ABC123321DEF

Output:

```
4 6
```

Notes

- Ensure the text files are located in the same directory as the program.
- This README was created by ChatGPT for spelling, formatting, and clarity and reviewed by student.