**2020BTECS00085, MANJIRI CHANDURE**

**CRYPTOGRAPHY AND NETWORK SECURITY LAB**

**ASSIGNMENT 2**

**Title: Encryption and Decryption using Transposition Cipher Technique.**

**1)**

**Aim: To Encrypt and decrypt plaintext using Rail Fence Cipher Technique**

**Theory:**

The Rail Fence Transposition Cipher, also known as the Zigzag Cipher, is a simple columnar transposition cipher technique. It involves arranging the plaintext characters in a zigzag pattern across multiple rows, known as "rails," and then reading them off row by row to create the encrypted message. While this cipher is easy to understand and implement, it lacks strong security and is mainly used for educational purposes or simple puzzles.

**Encryption:**

* Choose the number of rails (rows) for the zigzag pattern.
* Write the message diagonally across the rails, moving up and down.
* Read the characters row by row to form the encrypted message.

**Decryption:**

* Create the zigzag pattern with the chosen number of rails.
* Leave blank spaces in the pattern for characters to be placed.
* Fill in the blanks with the encrypted characters, row by row.
* Read the characters diagonally to retrieve the original message.

**Advantages**:

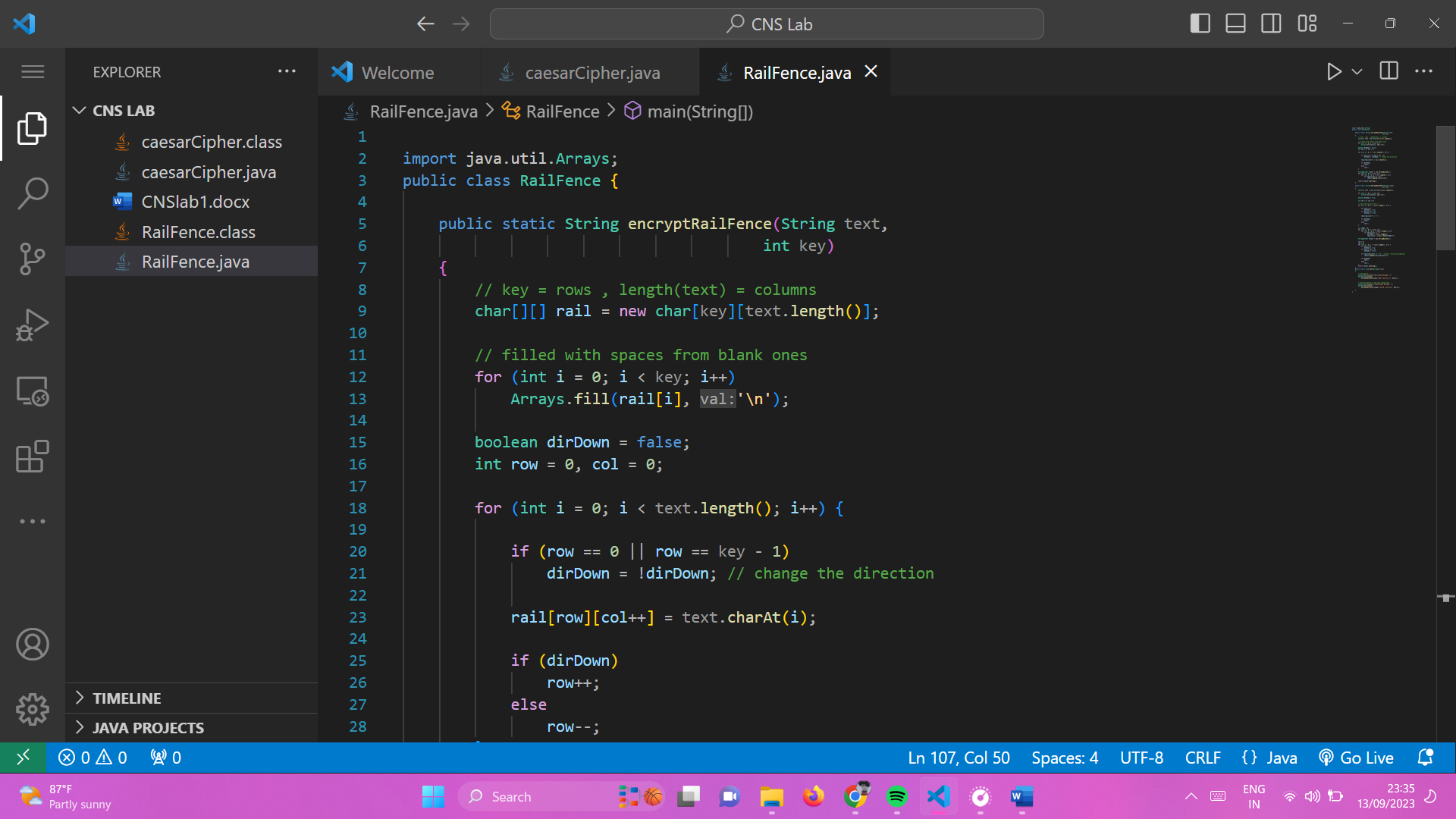
* Easy to understand and implement.
* Provides basic encryption and breaks up character repetition.

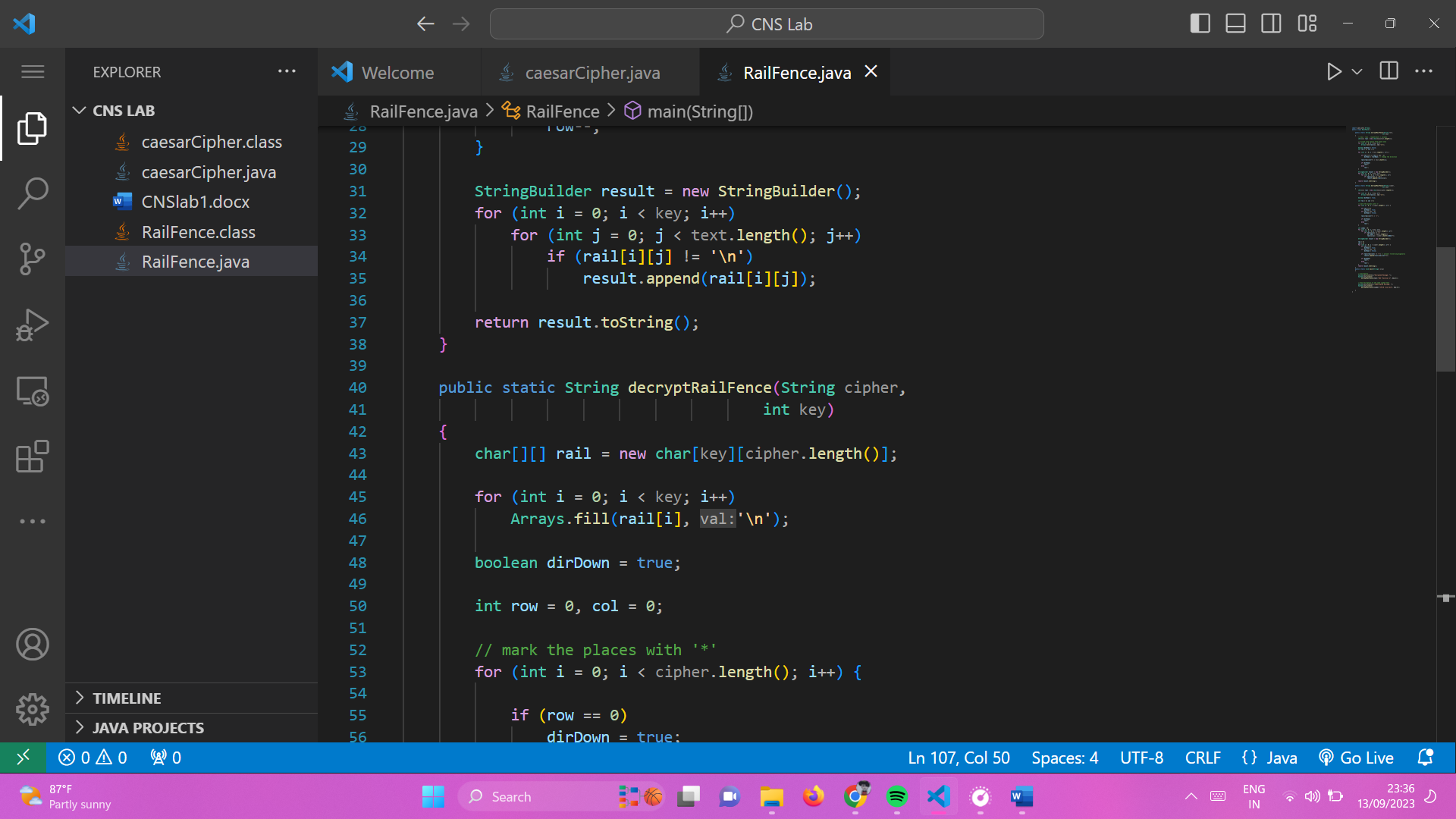
**Disadvantages:**

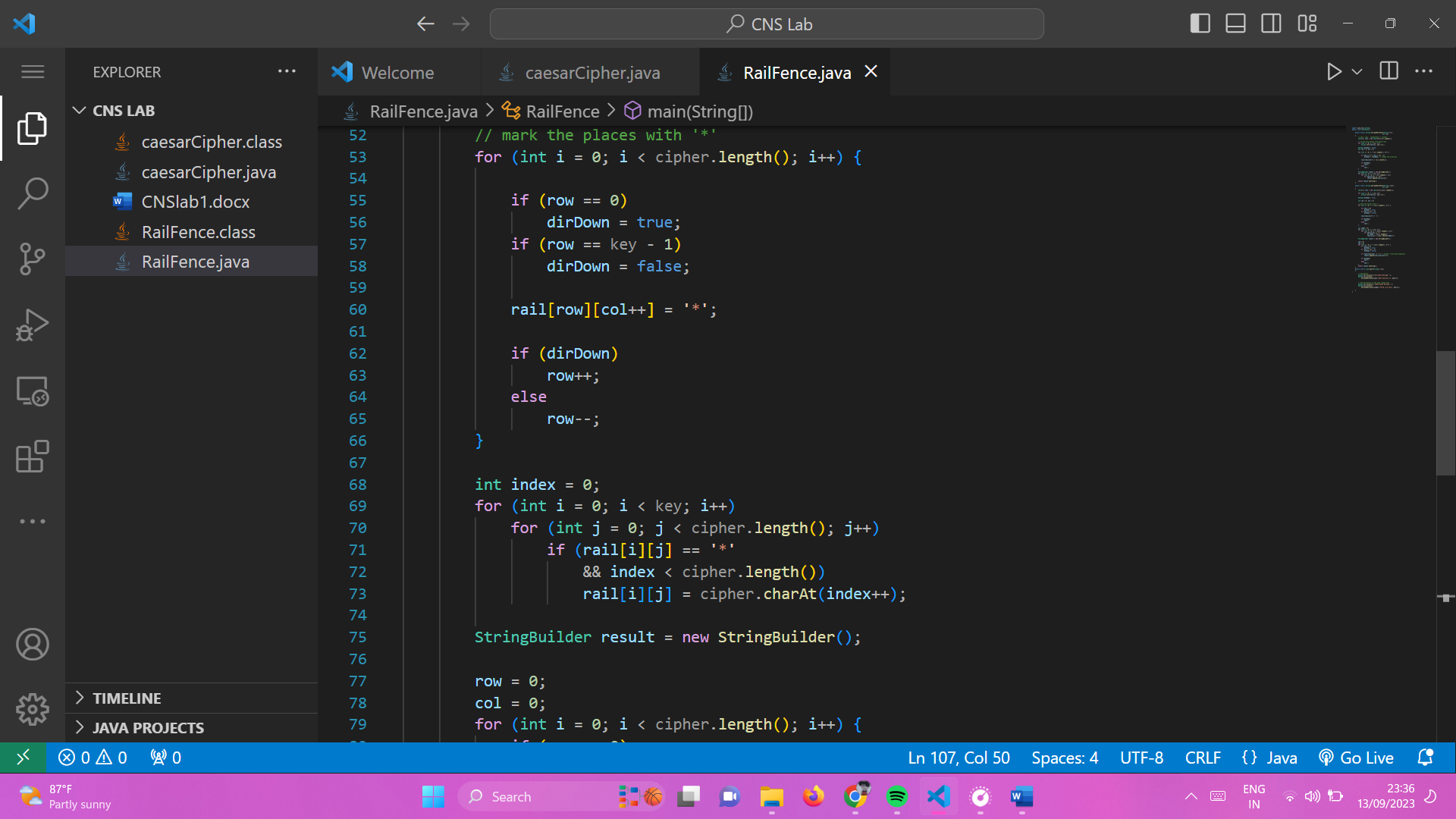
* Not secure against modern cryptanalysis.
* Security depends on the number of rails, making it less practical for strong encryption.

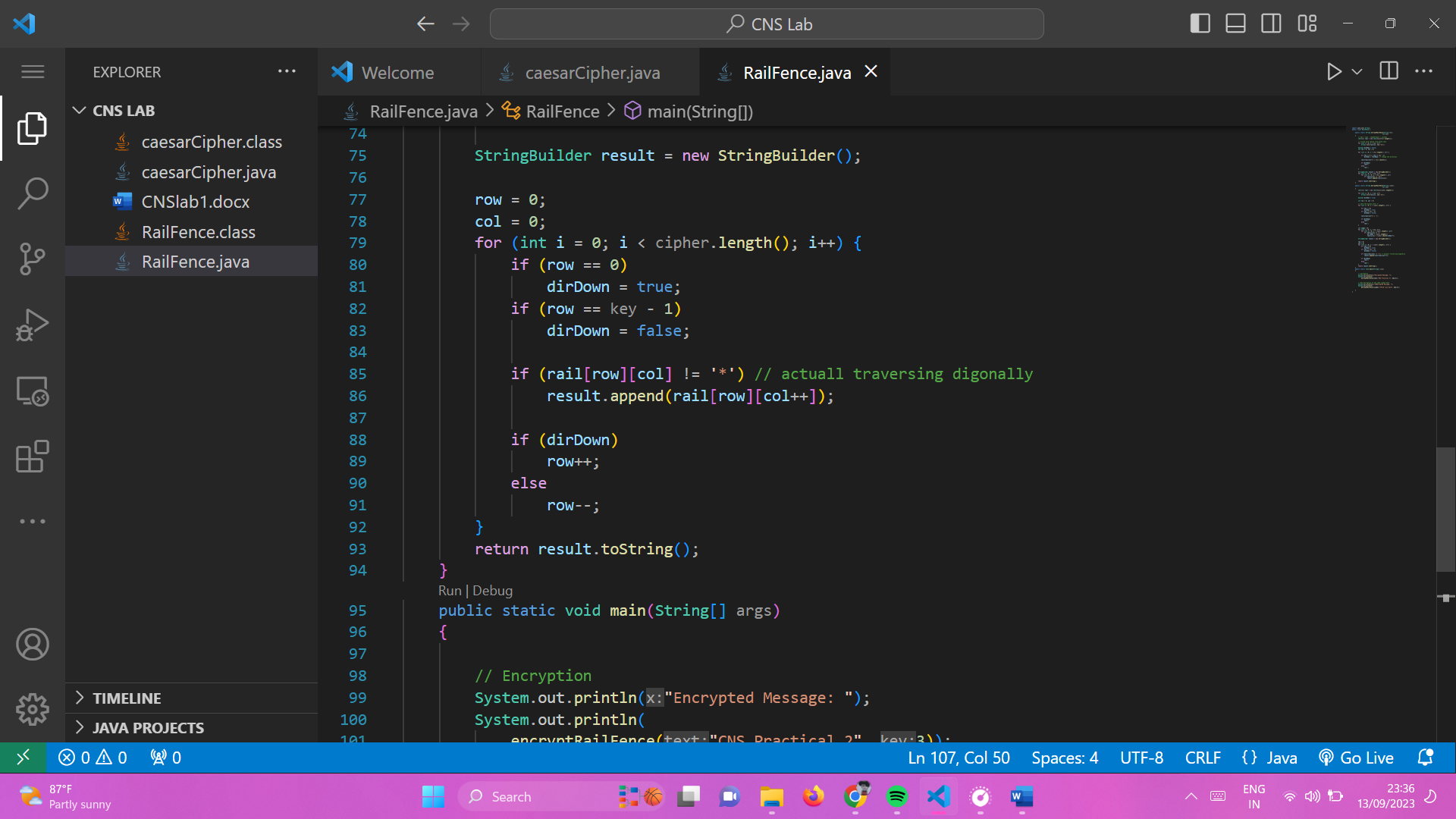
**Code:**

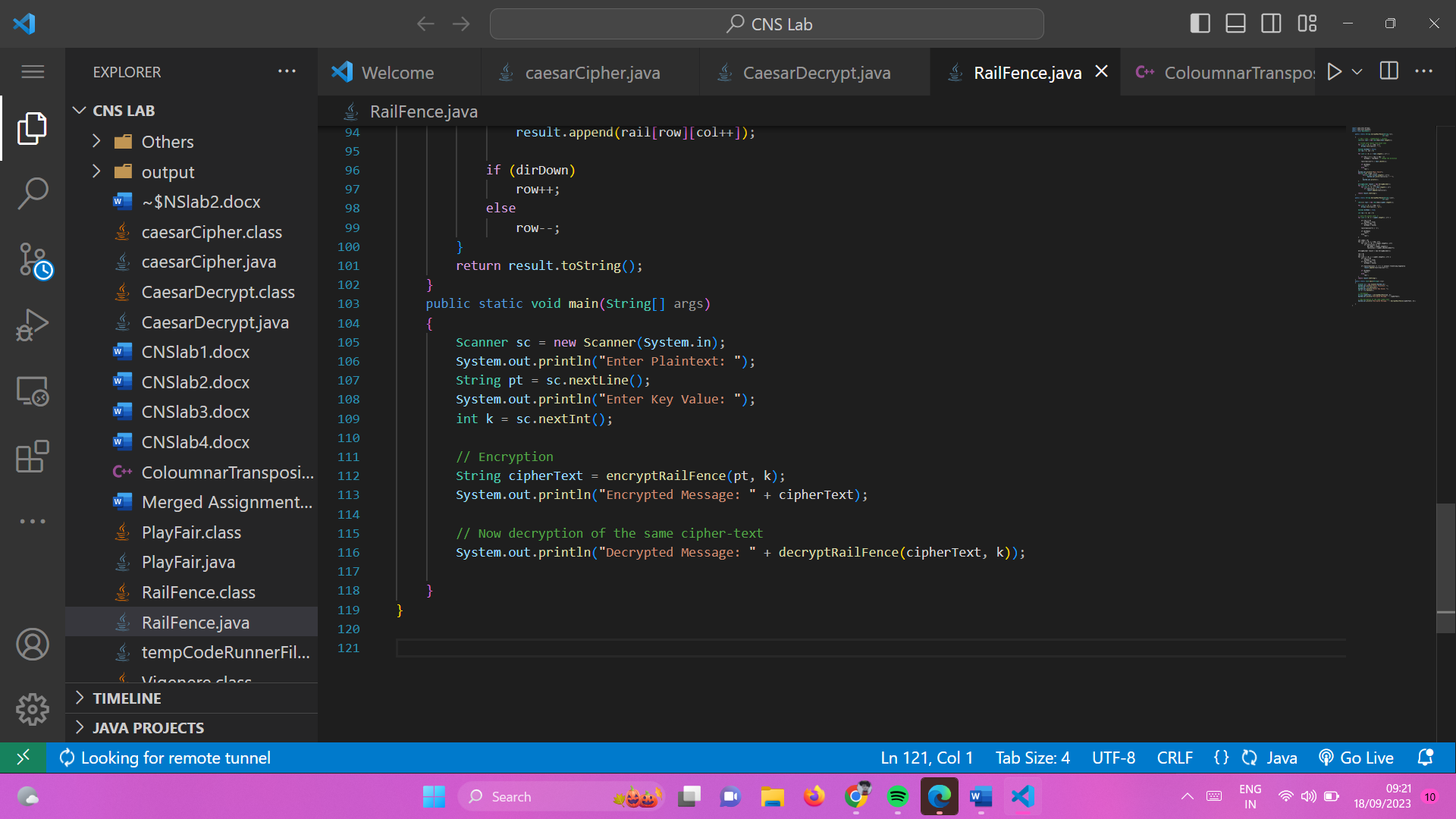
Implementation in Java



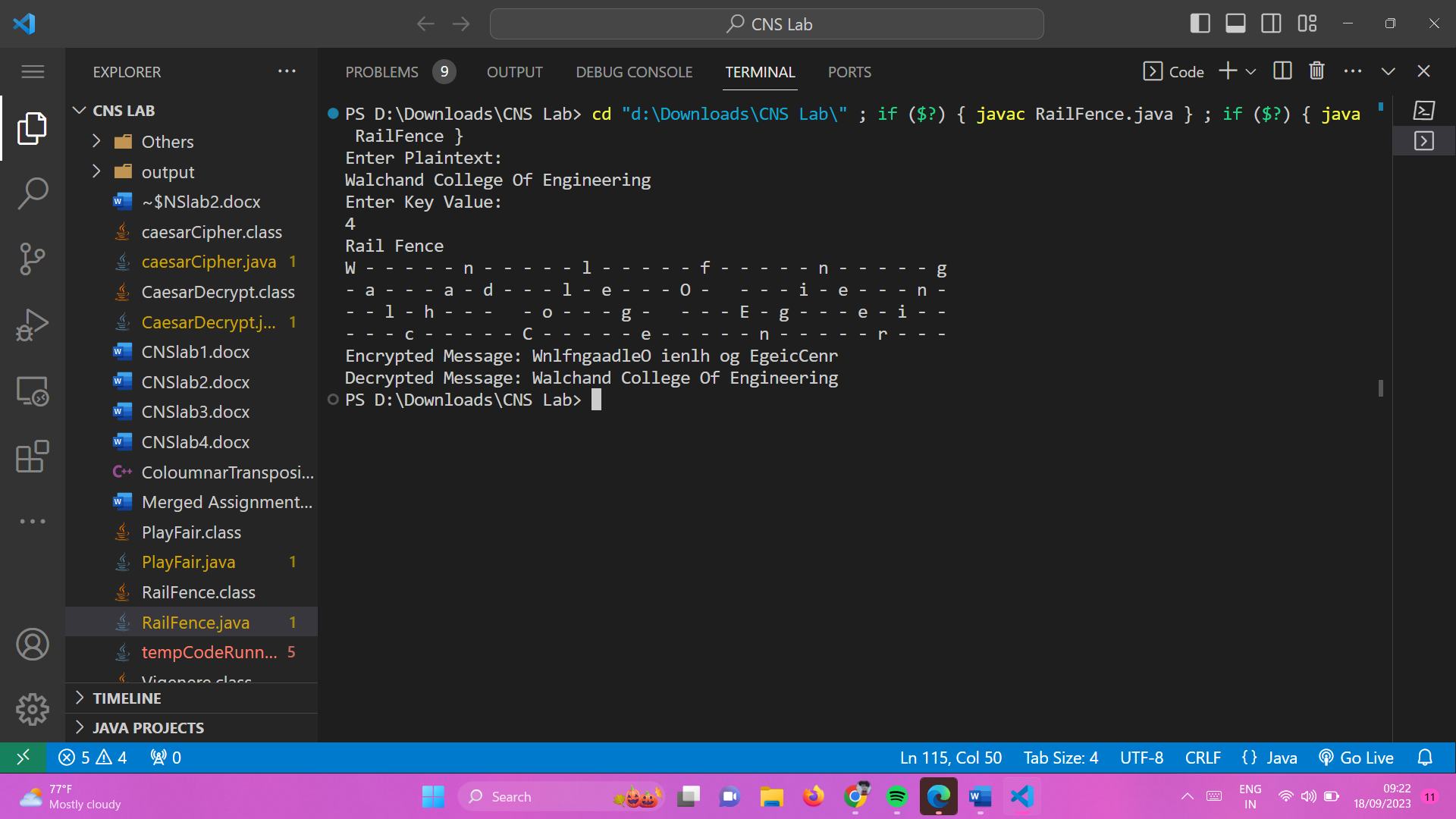








**Output:**



**2)**

**Aim: To encrypt and decrypt Columnar Transposition Cipher Technique**

**Theory:**

The Columnar Transposition Cipher is a more advanced transposition cipher technique that involves reordering the characters of a message based on a chosen keyword or key phrase.

It provides a higher level of security compared to simpler ciphers like the Rail Fence Cipher.

Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

**Encryption:**

* The message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order.
* Width of the rows and the permutation of the columns are usually defined by a keyword.
* For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be “3 1 2 4”.
* Any spare spaces are filled with nulls or left blank or placed by a character (Example: \_).
* Finally, the message is read off in columns, in the order specified by the keyword.

**Decryption:**

* Use the keyword to determine the order of columns in the transposition grid.
* Write the encrypted message into the grid column by column.
* Read the characters row by row to retrieve the original plaintext.

**Advantages:**

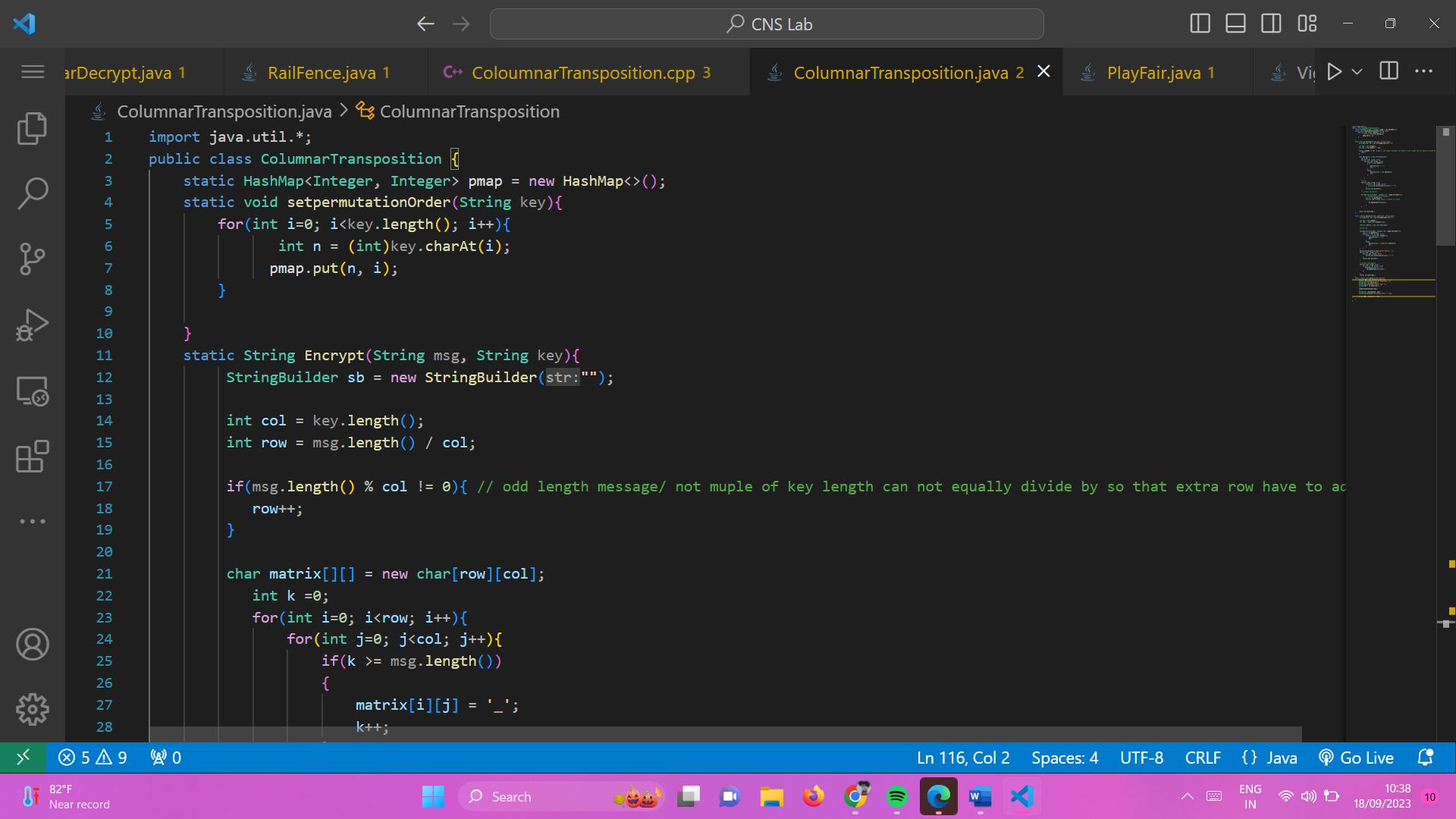
* Offers stronger security compared to simpler ciphers.
* Security depends on the length and uniqueness of the keyword.

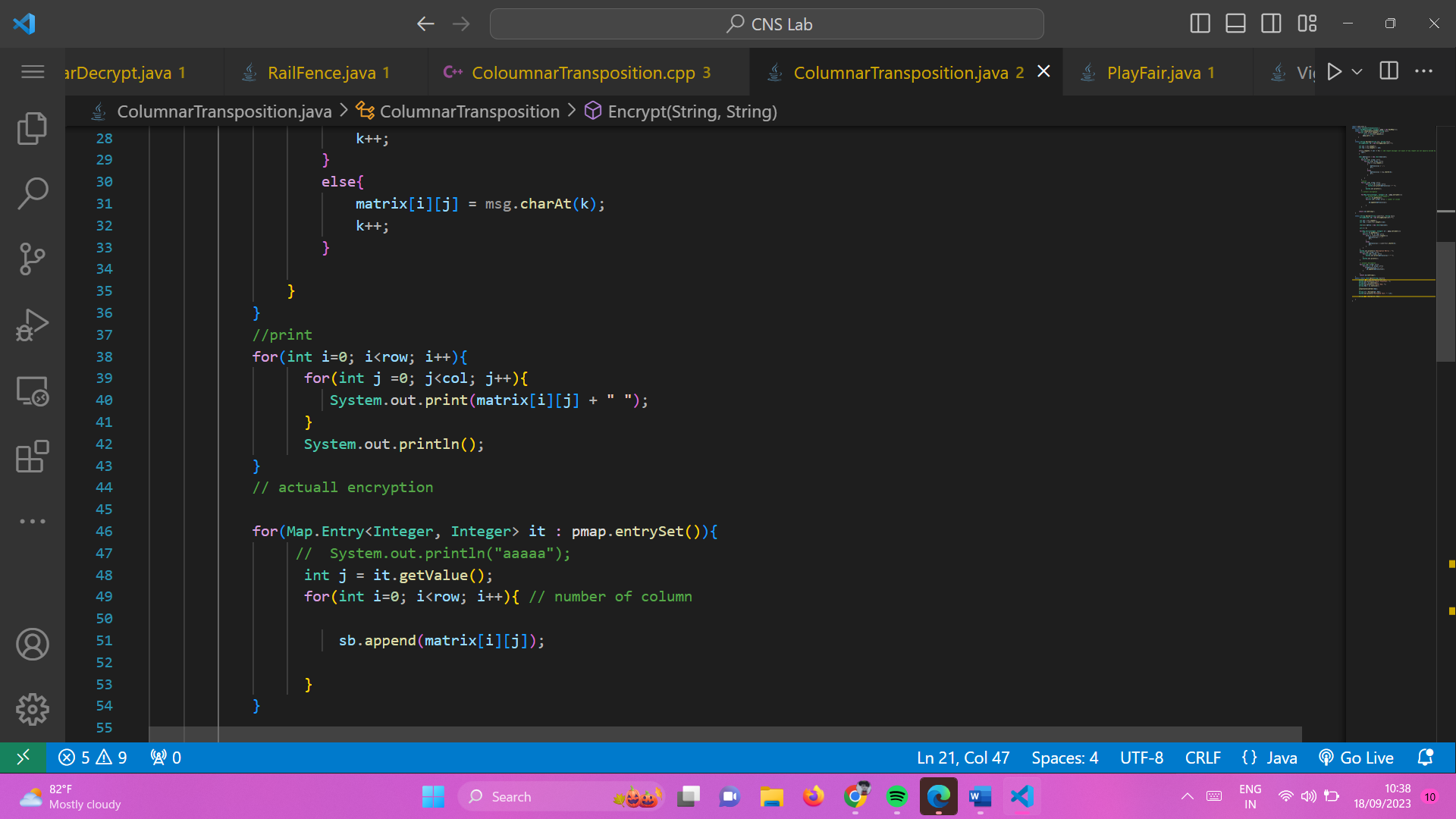
**Disadvantages**:

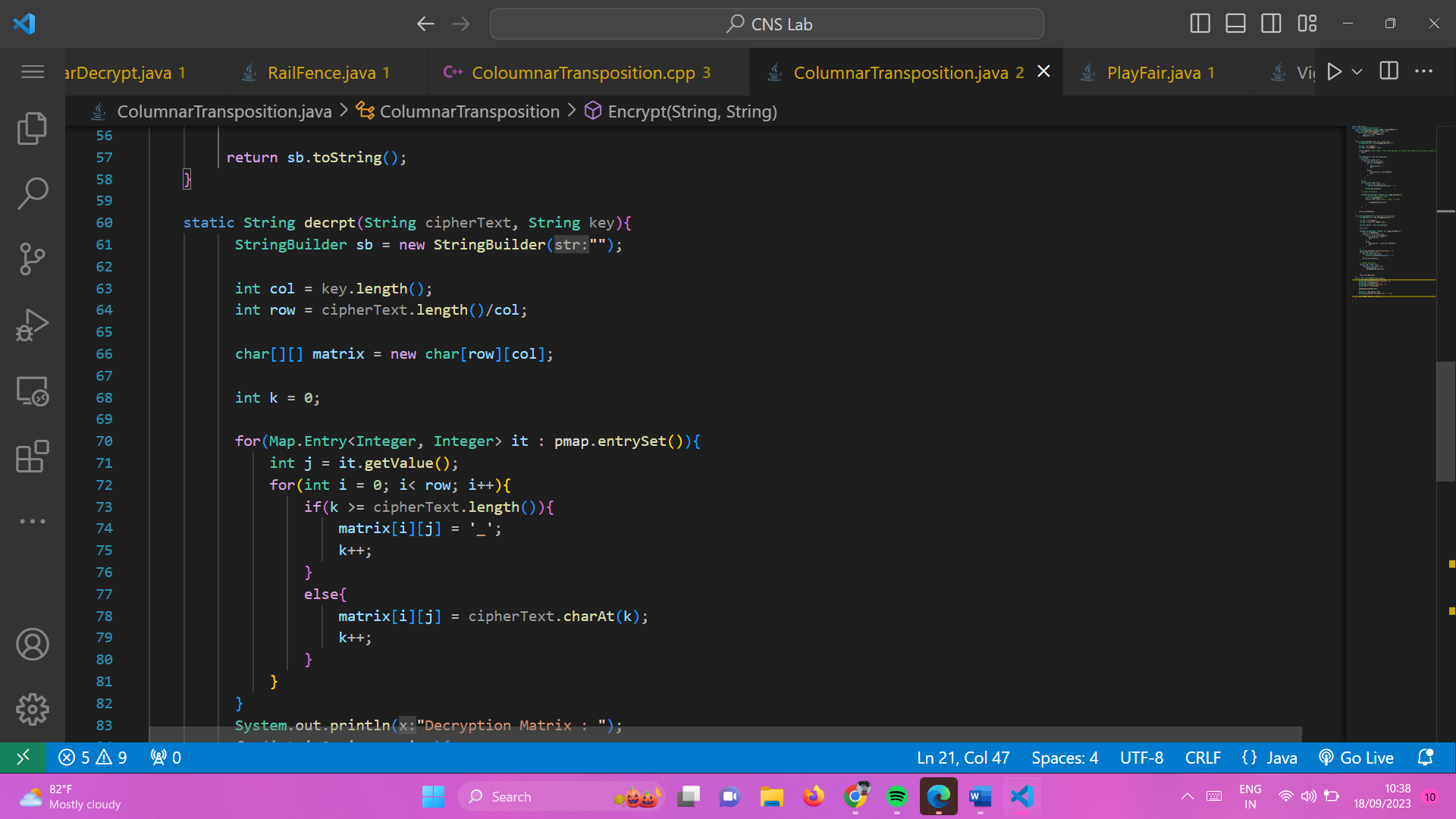
* Can be vulnerable to attacks if the keyword is short or easily guessed.
* May require additional padding characters for messages that don't fit evenly into the grid.

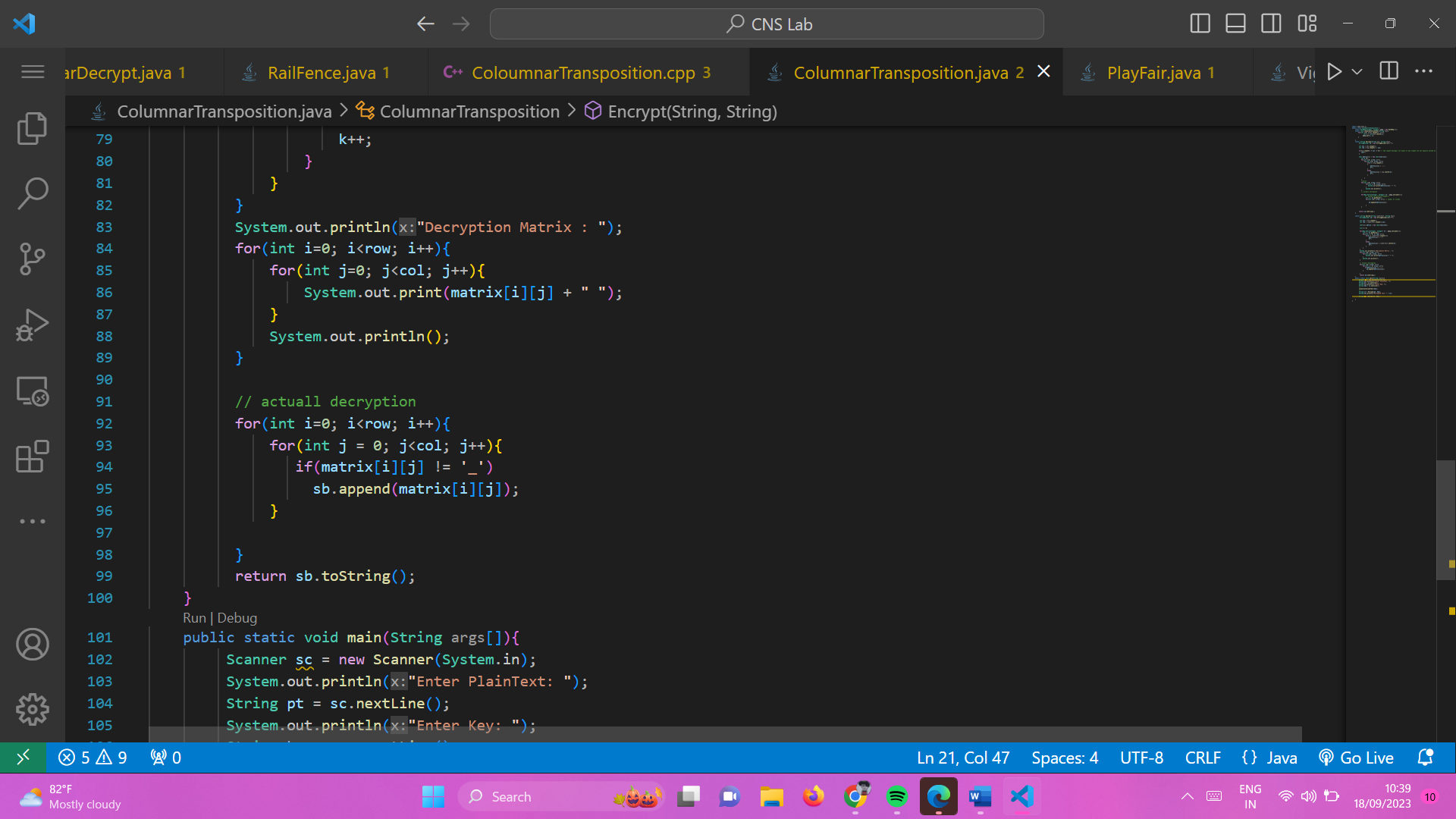
Implemented in java:

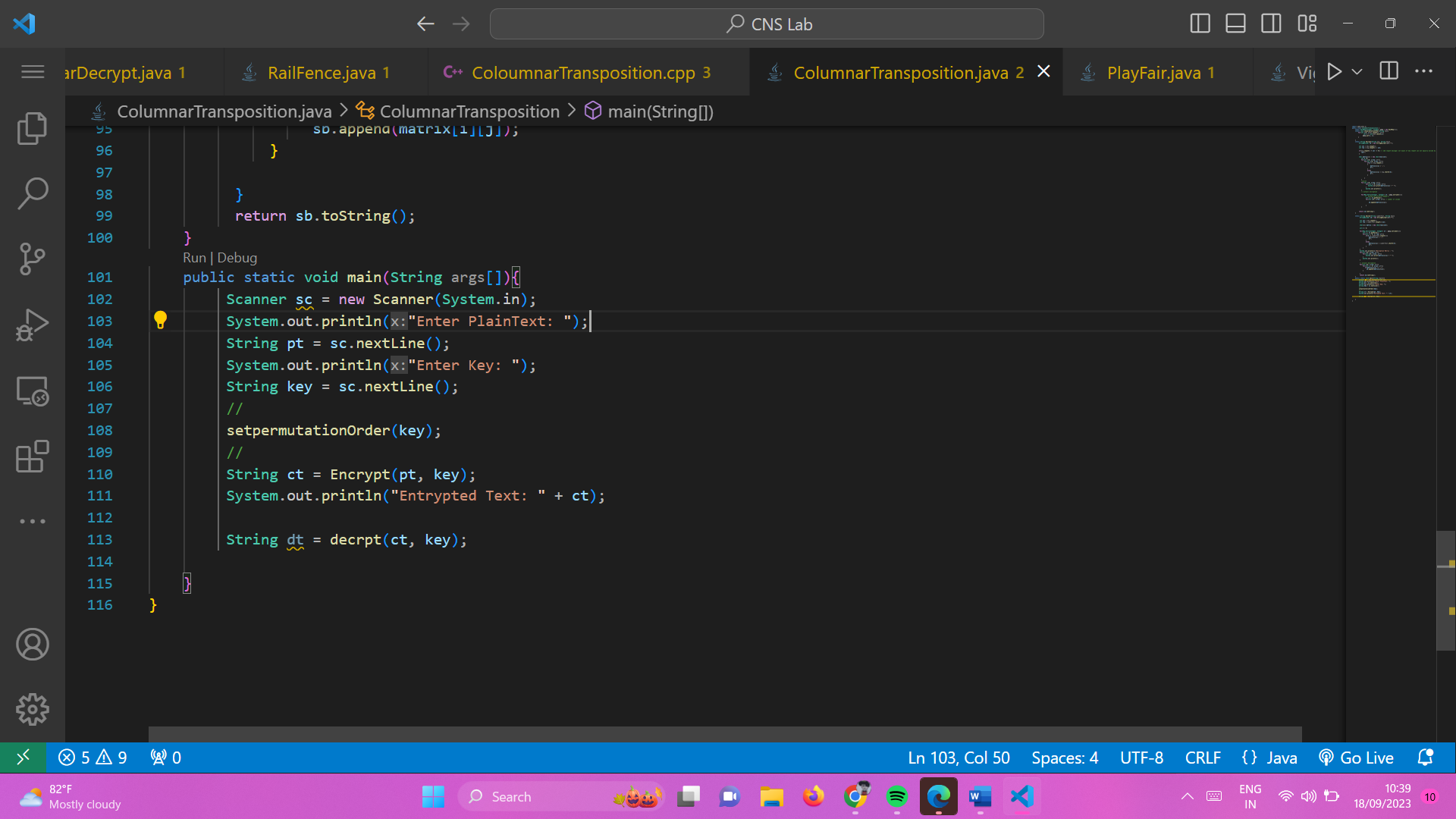
**Code:**











**Output:**

