

Ministry of Education and Culture of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 3

**Discipline**: Cryptography and Security

Elaborated: Șarov Andrei FAF-222,

Checked: Nirca Dumitru asist. univ.,

Chișinău 2024

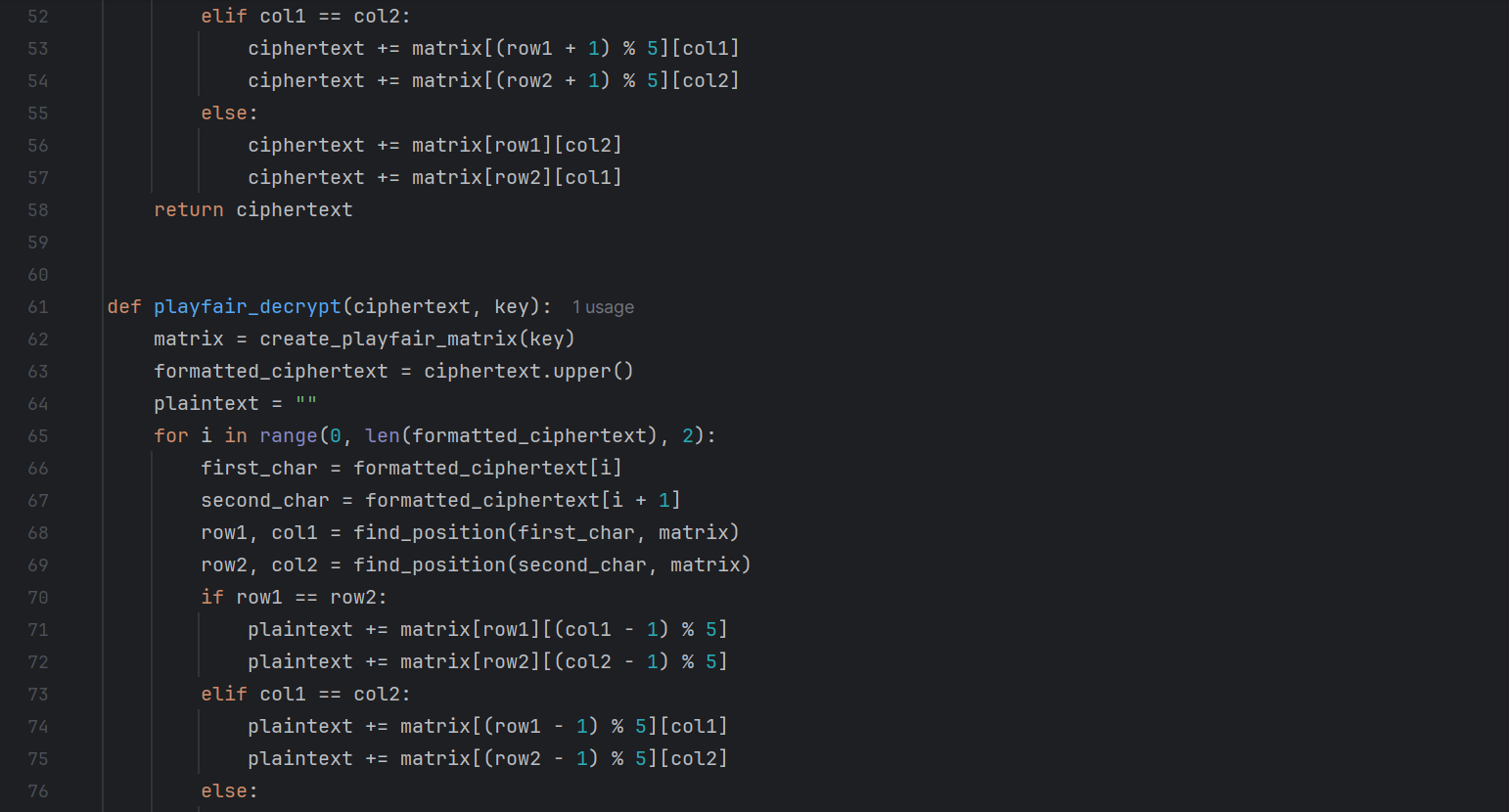
## Topic: Polyalphabetic Cipher

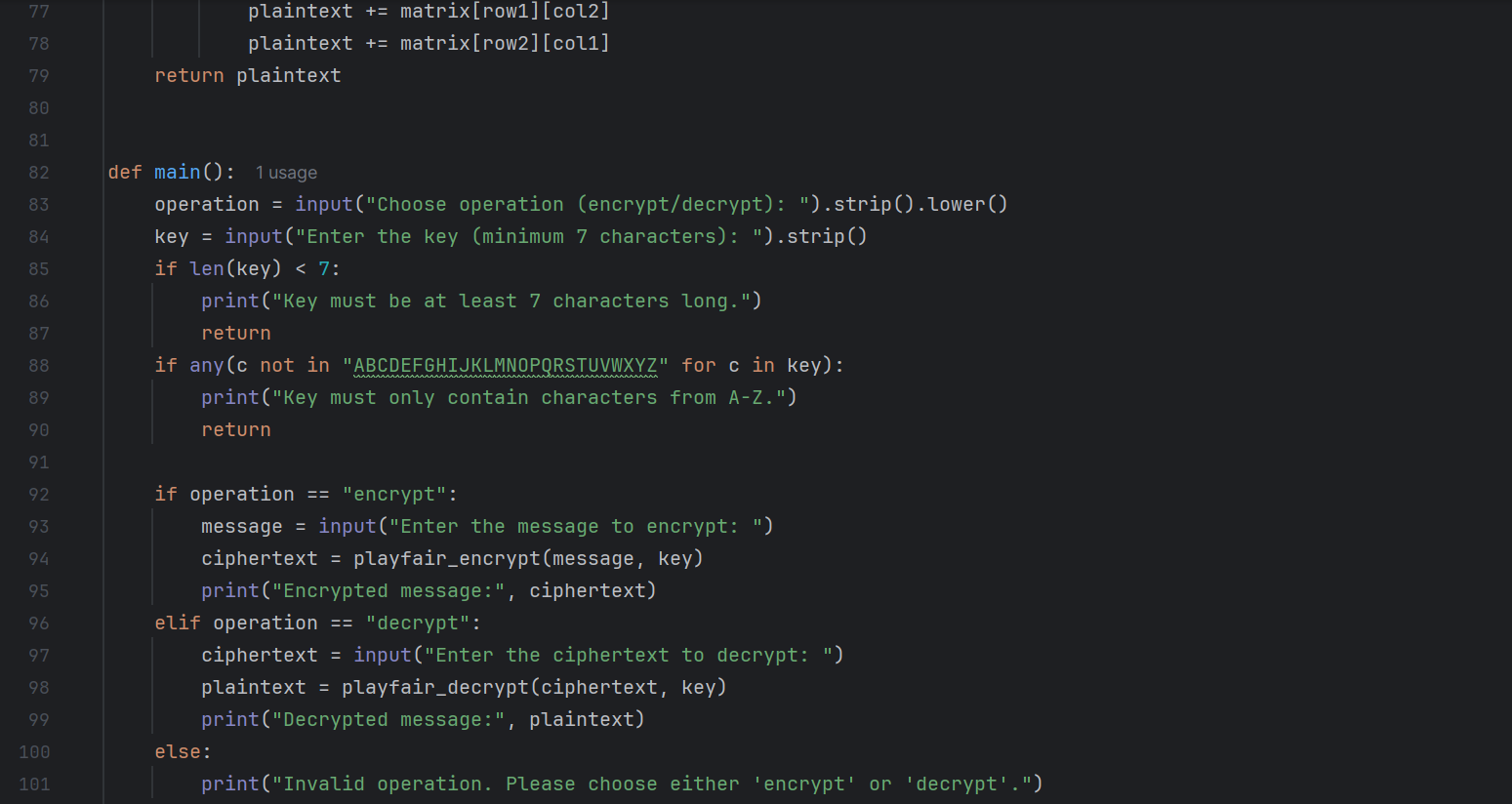
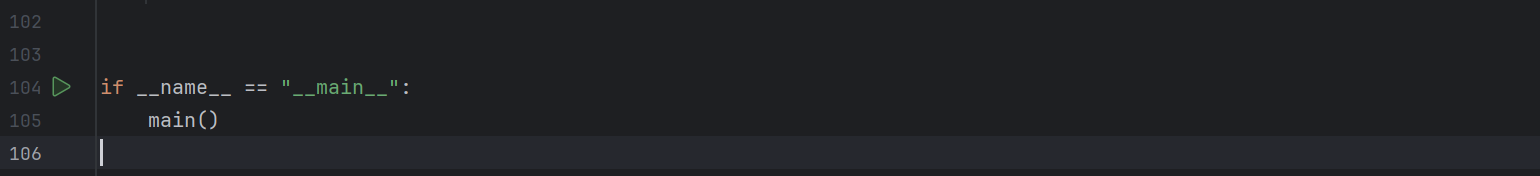
## Tasks:

1. Implement the Playfair algorithm in a programming language for messages in the Romanian language (31 letters). The values of the text characters range from ‘A’ to ’Z’, ’a’ to ’z’, and no other values are allowed. If the user enters other values, they will be suggested the correct range of characters. The length of the key must not be less than 7. The user will be able to choose the operation - encryption or decryption, will be able to enter the key, the message, or the ciphertext, and will obtain the ciphertext or the decrypted message. The final phase involves adding the...

## Theoretical notes:

Polyalphabetic Ciphers are a type of encryption that use multiple substitution alphabets to encrypt text. Unlike simple substitution ciphers, where each letter is replaced with another letter consistently, polyalphabetic ciphers use a key to determine which alphabet to use for each letter of the plaintext. The most famous example of a polyalphabetic cipher is the Vigenère cipher.

**Implementation (V1):  
  
 ** **** 

** **

1. The implementation begins by creating a unique key matrix derived from the user-provided key. The key is processed by removing duplicates and converting it to uppercase while treating 'J' as 'I' since the Playfair cipher uses a 25-letter matrix. The key matrix is then constructed by appending the remaining letters of the alphabet, ensuring no letters are repeated and that 'J' is excluded.
2. The next step involves formatting the input message to ensure it is compatible with the Playfair cipher rules. The message is converted to uppercase, with 'J' replaced by 'I'. Any duplicate letters within a pair are separated by inserting an 'X'. Additionally, if the final formatted message contains an odd number of characters, an 'X' is appended at the end to create a complete pair.
3. During the encryption process, the formatted message is processed in pairs of letters. The positions of each letter in the key matrix are determined. If both letters are located in the same row, they are replaced by the letters immediately to their right. If they are in the same column, they are substituted with the letters directly below them. If the letters form a rectangle in the matrix, each letter is replaced by the letter that occupies its row and the column of the other letter.
4. The decryption process mirrors the encryption but with a key distinction: letters are replaced by the letters to their left or above them based on their positions in the matrix. This allows for the retrieval of the original message from the ciphertext.
5. The user is prompted to select whether they wish to encrypt or decrypt a message. They provide the key and the respective message or ciphertext. The program checks the length of the key, ensuring it meets the minimum requirement of seven characters. Additionally, it verifies that the key contains only valid alphabetic characters. Finally, the program outputs either the encrypted message or the decrypted plaintext based on the user's choice.

**Conclusions:**

In this laboratory task, we implemented the Playfair cipher, a polyalphabetic encryption technique suitable for the Romanian language. By creating a flexible and user-friendly Python program, we demonstrated the principles of this encryption method while ensuring adherence to input constraints. This exercise highlights the importance of secure communication methods and enhances understanding of classical cryptography concepts.