

Ministry of Education and Culture of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 2

**Discipline**: Cryptography and Security

Elaborated: Șarov Andrei FAF-222,

Checked: Nirca Dumitru asist. univ.,

Chișinău 2024

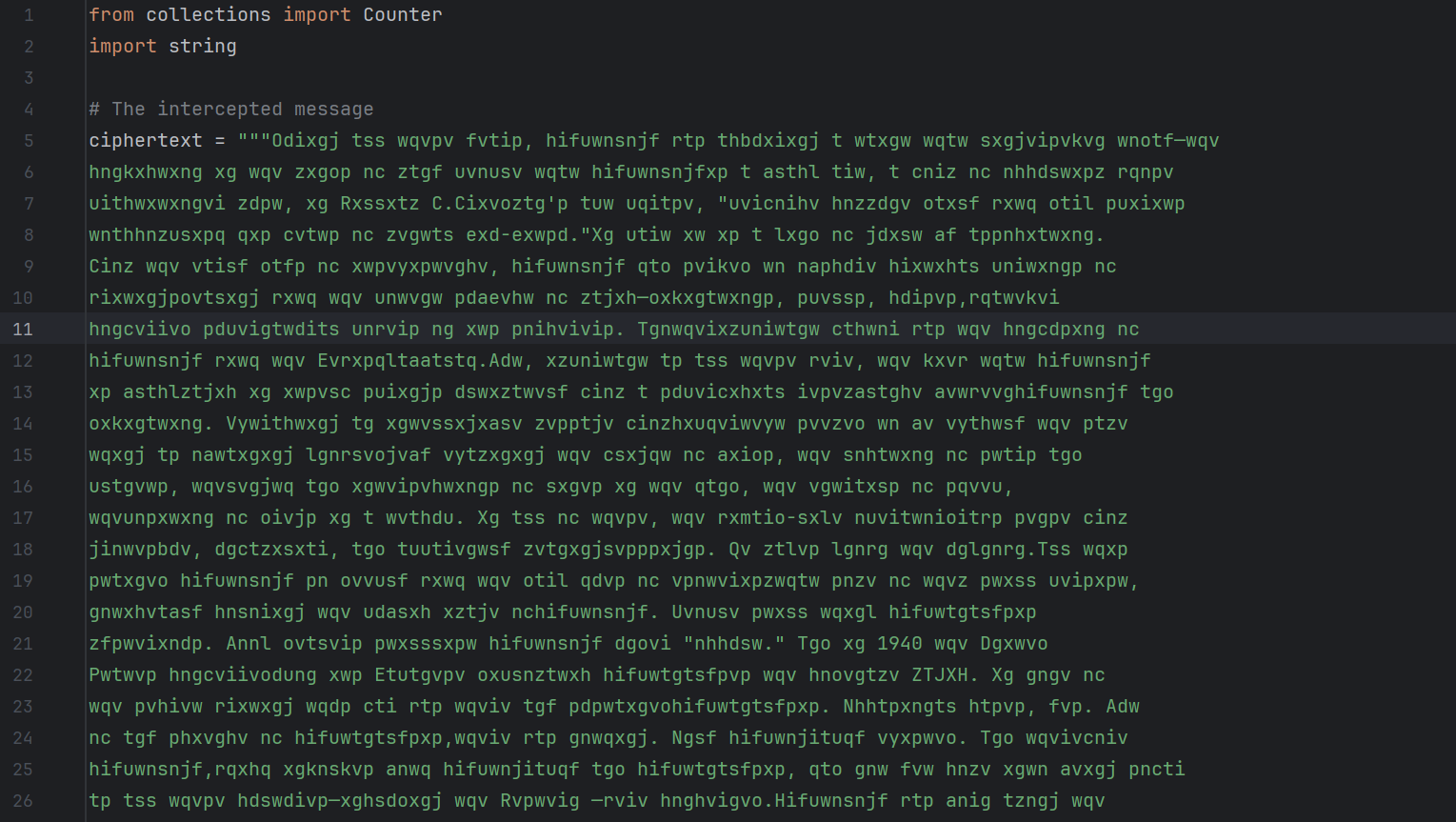
## Topic: Oryptanalysis of Monoalphabetic ciphers

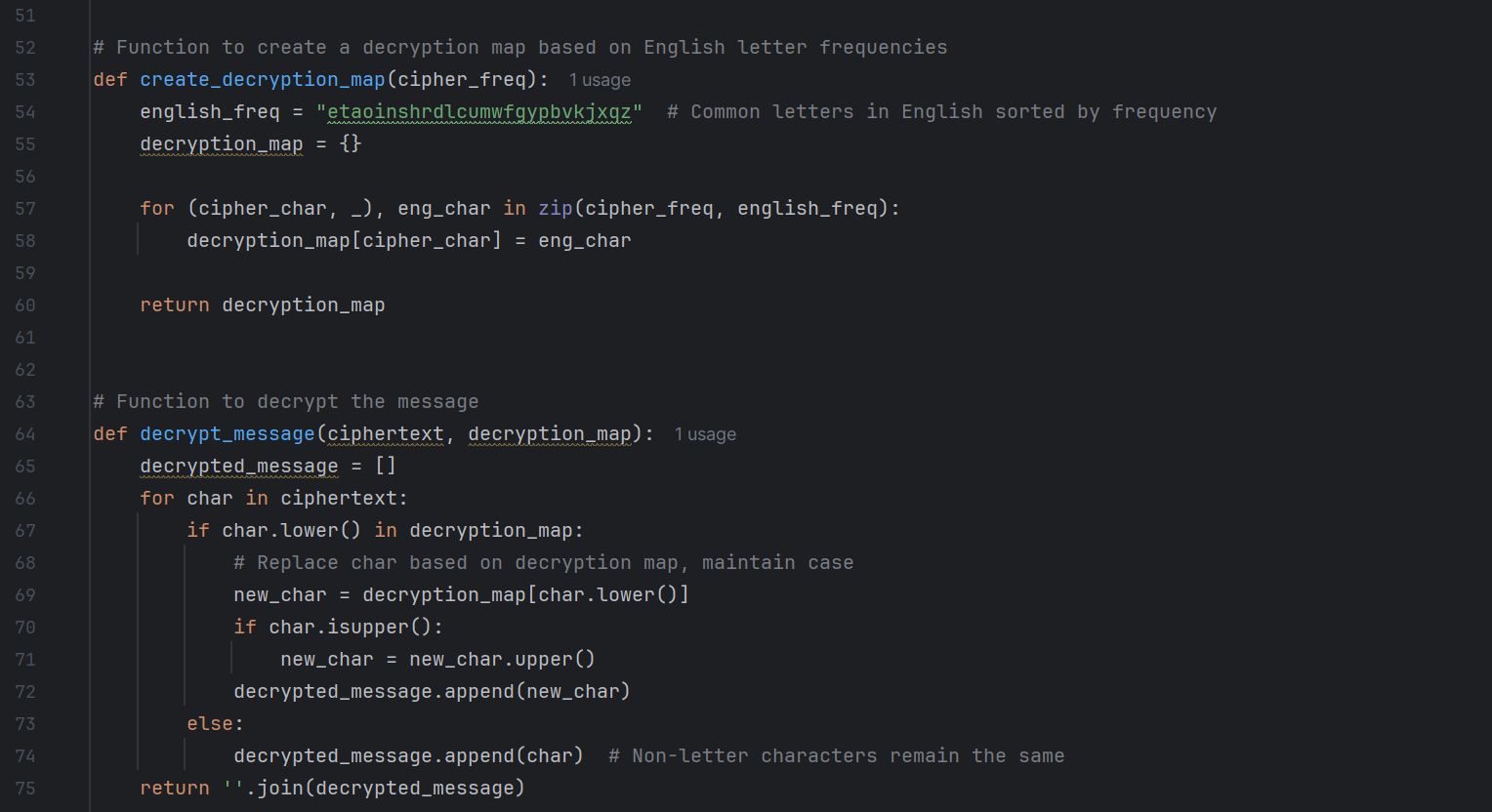
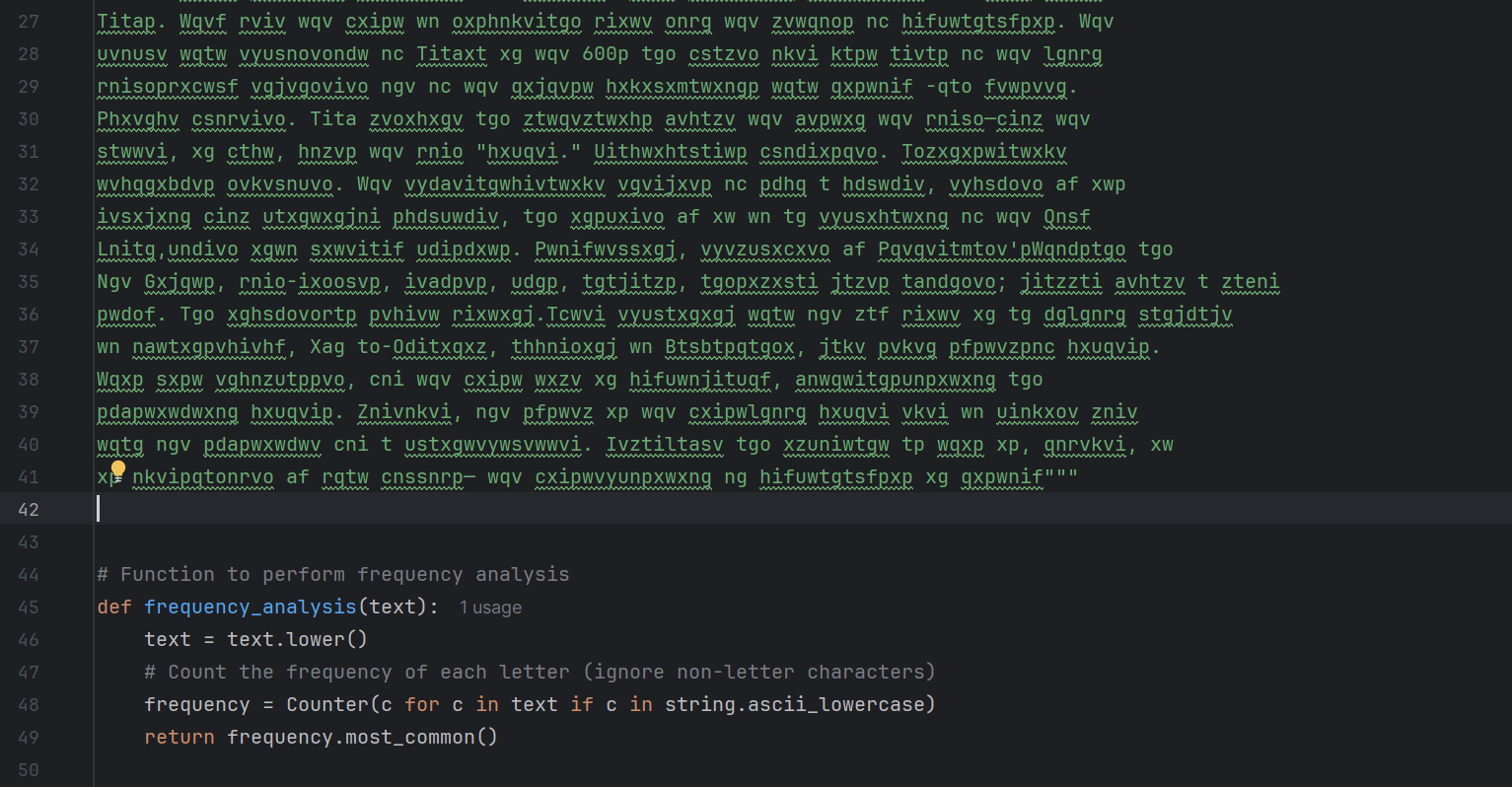
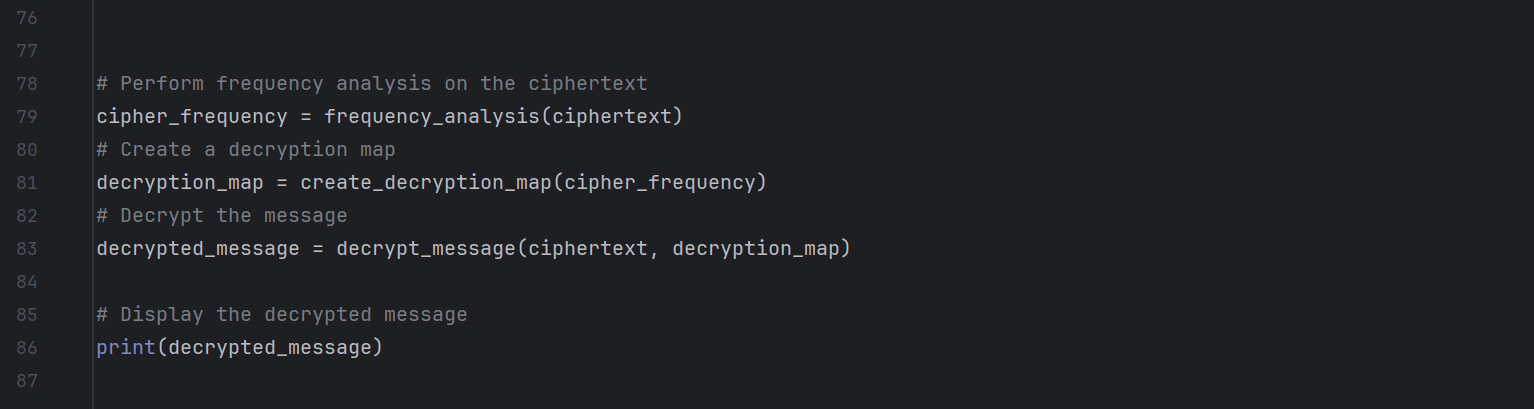
## Tasks:

1. A message was intercepted that is known to have been obtained using a monoalphabetic cipher. By applying frequency analysis, determine the original message, assuming it is a text written in English. Note that only the letters were encrypted, with other characters remaining unencrypted.

## Theoretical notes:

Monoalphabetic ciphers substitute each letter of the plaintext with a fixed, unique letter of the ciphertext alphabet. Frequency analysis, a standard cryptanalysis technique for such ciphers, involves comparing the frequency of each character in the encrypted text with the typical frequency distribution of letters in the target language. Since English text exhibits predictable letter patterns, the high-frequency letters in the ciphertext often correspond to high-frequency letters in English. This technique is effective due to the structural regularities in English and provides a foundation for breaking monoalphabetic ciphers.

**Implementation (V4):  
  
 **

** **

1. The code begins by importing necessary libraries: `Counter` from `collections` for counting letter frequencies and `string` for accessing English letters. It then defines the `ciphertext`, which contains the encrypted message.
2. A function called `frequency\_analysis` is created to analyze the letter frequencies in the ciphertext. This function converts the text to lowercase, counts the occurrences of each letter using `Counter`, and returns a sorted list of the most common letters.
3. Next, the `create\_decryption\_map` function is defined. It establishes a mapping between letters in the ciphertext and the most common letters in English. This is done by pairing the results from the frequency analysis with a predefined string of English letters ordered by frequency, resulting in a dictionary that serves as the decryption map.
4. The `decrypt\_message` function then uses this map to decrypt the ciphertext. It initializes an empty list for the decrypted characters, iterates through each character in the ciphertext, and replaces letters based on the decryption map while maintaining the original case. Non-letter characters are added unchanged.
5. After defining these functions, the code executes the frequency analysis on the ciphertext and creates the decryption map. It then decrypts the message using the defined functions and prints the final decrypted text to the console.
6. This explanation covers the flow and logic of the code without excessive bullet points, providing a clear understanding of its structure and functionality.

**Conclusions:**

The use of frequency analysis facilitated deciphering the monoalphabetic cipher text by correlating the letter frequencies with typical English patterns. The output demonstrates a partially or fully deciphered message, depending on the complexity of the substitutions used. This method highlights the weaknesses in simple substitution ciphers, underlining the importance of polyalphabetic or more complex ciphers for stronger encryption.