Ministerul Educaţiei şi Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică şi Microelectronică

Laboratory work 3

Subject: Polyalphabetic script encryption

Done by: Gitlan Gabriel
st. gr. FAF-213

Verified by: Mîţu Cătălin
asist. univ.

Chişinău - 2023

Description:
The encryption method known as the "Vigenère cipher" was mistakenly attributed to Blaise de Vigenère in the 19th century and was actually first described by Giovan Battista Bellaso in his 1553 book "La cifra del." Mr. Vigenère created a similar but still different and more powerful cipher in 1586.

Task:
To implement the Vigenere algorithm in one of the programming languages for messages in Romanian (31 letters), which are encoded with the numbers 0, 1, ... 30. The values of text characters are between 'A' and 'Z', 'a' and 'z', and no other values are allowed. In case the user enters other values, the correct range of characters will be suggested. The key length must not be less than 7. Encryption and decryption will be carried out according to the formulas in the mathematical model presented above. In the message, spaces must be removed first, then all letters will be transformed into uppercase. The user will be able to choose the operation - encryption or decryption, input the key, the message, or the ciphertext, and obtain the ciphertext or the decrypted message.

```java
public static String encrypt(String plaintext, String key) {
    check(key);
    List<Character> plainList = plaintext.toUpperCase().replaceAll(" ", "").chars().mapToObj(i -> (char) i)
            .toList();
    List<Character> keyList = key.toUpperCase().replaceAll(" ", "").chars().mapToObj(i -> (char) i).toList();
    List<Character> keyFullList = new ArrayList<>();
    int i = 0;
    int j;
    int k = 1;
    while (i < plainList.size()) {
        j = i - (keyList.size() * (k - 1));
        keyFullList.add(keyList.get(j));
        if ((i + 1) / k == keyList.size()) {
            k++;
        }
        i++;
    }
    List<Integer> plainPosition = plainList.stream().map(character -> getAlphabet().indexOf(character)).toList();
    List<Integer> keyFullPosition = keyFullList.stream().map(character -> getAlphabet().indexOf(character))
            .toList();
    List<Integer> difference = new ArrayList<>();

    int s = 0;
    while (s < plainPosition.size()) {
        difference.add((plainPosition.get(s) + keyFullPosition.get(s)) % getAlphabet().size());
        s++;
    }
    return difference.stream()
            .map(integer -> getAlphabet().get(integer))
            .map(String::valueOf)
            .collect(Collectors.joining());
}
```

Figure 1: Text Encryption

```java
public static String decrypt(String ciphertext, String key) {
    check(key);
    StringBuilder plaintext = new StringBuilder();
    ciphertext = ciphertext.toUpperCase();
    key = key.toUpperCase();

    for (int i = 0, j = 0; i < ciphertext.length(); i++) {
        char currentChar = ciphertext.charAt(i);
        if (getAlphabet().contains(currentChar)) {
            int shift = getAlphabet().indexOf(key.charAt(j % key.length()));
            char decryptedChar = getAlphabet().get(
                    (getAlphabet().indexOf(currentChar) - shift + getAlphabet().size()) % getAlphabet().size());
            plaintext.append(decryptedChar);
            j++;
        } else {
            plaintext.append(currentChar);
        }
    }

    return plaintext.toString();
}
```

Figure 2: Text decryption

```java
public static void main(String[] args) {
    String plaintext = "Mișa s-a sfădit cu țiganul Vaniușa";
    String key = "Cătălin";

    String encryptedText = encrypt(plaintext, key);
    String decryptedText = decrypt(encryptedText, key);

    System.out.println("Original Text:  " + plaintext);
    System.out.println("Encrypted Text: " + encryptedText);
    System.out.println("Decrypted Text: " + decryptedText);
}
```

Figure 3: main function

```
┌─[Dell-5540🔥gaby]─[⊶main]─[~/.../Lab3]
└─>>> java Main.java
Original Text:  Mișa s-a sfădit cu țiganul Vaniușa
Encrypted Text: QÎLĂCHNUGȚEȚÂRYUÂHLVIPWTOȚCFC
Decrypted Text: MIȘASZASFĂDITCUȚIGANULVANIUȘA
┌─[Dell-5540🔥gaby]─[⊶main]─[~/.../Lab3]
└─>>>
```

Figure 4: Output

Conclusion:

At the current laboratory work I studied how polyalphabetic decryption works and how to implement Vigenere algorithm. It was a real challenge to do it for the Romanian language but it worked. After that, I had found that the text before encryption and that one after decryption is the same.

Link to repo: https://github.com/5anji/CS_Labs