Ministry of Education of Republic of Moldova

Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Software Engineering Department

Cryptography

Laboratory work #1

Cifrul Caesar

Author:
Chirtoaca Liviu
std. gr. FAF-233

Verified:
Zaica M.

Chișinău 2025

Purpose of Laboratory Work

The purpose of this laboratory work is to implement the Caesar cipher algorithm for encrypting and decrypting text using both simple numeric keys and keyword-based substitution alphabets. This work aims to understand basic cryptographic principles and develop programming skills in Python.

Conditions of the Problem

1. Implement basic Caesar cipher with numeric key (1-25) for encryption and decryption
2. Develop keyword-based Caesar cipher that uses custom alphabets derived from user keywords
3. Preserve character case and handle non-alphabetic characters appropriately in keyword cipher

Technical Implementation

The implementation consists of two main cipher variants:

Simple Caesar Cipher: Uses numeric keys to shift letters in the standard alphabet. Text is converted to uppercase and spaces are removed in the output format.

Keyword Caesar Cipher: Creates custom alphabets by removing duplicate letters from user keywords and appending remaining alphabet letters. This cipher preserves original character case and maintains non-alphabetic characters in their positions.

The program provides a menu-driven interface allowing users to encrypt, decrypt, and analyze cryptograms using both cipher methods. Input validation ensures keywords are minimum 7 unique characters and contain only letters, while numeric keys are restricted to the range 1-25.

Key functions include character shifting with modulo arithmetic for alphabet wrapping, duplicate removal from keywords, custom alphabet generation, and comprehensive cryptogram analysis that displays all possible decryptions.

```
def Process_text(text):
    text=text.upper()
    proccesed_text = ''.join([char for char in text if char
       in Alphabet])
    return proccesed_text
```

This part of the code removes all the spaces the text have and makes all the letters to uppercase for easier use

```
def Validate_text(text):
    text = text.upper()
    for char in text:
        if char not in Alphabet and char != '␣':
            if char in '!?.,:;-_()[]{}"\'' :
                continue
            print(f"Invalid␣character:␣'{char}'.␣Text␣must␣
               contain␣only␣A-Z,␣spaces,␣and␣basic␣
               punctuation.")
            return False
    return True
```

This function helps validating the text to be sent for being processed it skips the special charachters not to be included

```
def Permuted_alpabet_cipher(key2):
    key2 = key2.upper()
    permuted= ""
    for char in key2:
        if char not in permuted and char in Alphabet:
            permuted += char
    for char in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ':
        if char not in permuted:
            permuted += char
    return permuted
```

This function creates the new alphabet by introducing the key2 to the alphabet removing duplicates and then introducing the remaining letters in normal order

```
==================================================
           CIPHER ENCRYPTION/DECRYPTION TOOL
==================================================
1. Caesar Cipher
2. Permutation Cipher
3. Exit
==================================================
Enter your choice (1-3): 1

--- CAESAR CIPHER ---
Enter mode (encrypt/decrypt): encrypt
Enter key (1-25): 3
Enter text: hello
Encrypted text: KHOOR

Press Enter to continue...

==================================================
           CIPHER ENCRYPTION/DECRYPTION TOOL
==================================================
1. Caesar Cipher
2. Permutation Cipher
3. Exit
==================================================
Enter your choice (1-3): 2

--- PERMUTATION CIPHER ---
Enter mode (encrypt/decrypt): encrypt
Enter key (1-25): 3
Enter permutation key (string of letters A-Z): cryptography
Enter text: hello
Encrypted text: EJQQH

Press Enter to continue...
```

Figure 1: Console

Conclusions

This laboratory work showed the implementation of two Caesar cipher variants in Python: the numeric-shift cipher and the keyword-based substitution cipher. The simple version demonstrated the basic principle of substitution using modular arithmetic, while the keyword version added the ability to generate custom alphabets from user input.

The program handled case sensitivity, preserved non-alphabetic characters, and included input validation to ensure correct data processing. These features made the ciphers more reliable and practical to use.

The work also highlighted the limits of classical ciphers, which are easy to break with brute force or frequency analysis, but remain useful for learning the fundamentals of cryptography. This lab provided both programming practice and a better understanding of substitution ciphers.