Laboratory work 2- Caesar Cipher Encryption and Decryption

Author: Dávid Kromka
Personal key: 11

This Python script implements the Caesar cipher encryption and decryption algorithms. The Caesar cipher is a substitution cipher where each letter in the plaintext is shifted a fixed number of positions down or up the alphabet.

Functions:

1. **caesar_encrypt(plaintext, key)**:
    - **Input: plaintext** (string) - The message to be encrypted, **key** (integer) - The shift value.
    - **Output:** Encrypted text (string).
    - **Description:** Encrypts the input **plaintext** using the Caesar cipher with the given **key**. Non-alphabetic characters remain unchanged.
2. **caesar_decrypt(ciphertext, key)**:
    - **Input: ciphertext** (string) - The encrypted message to be decrypted, **key** (integer) - The shift value.
    - **Output:** Decrypted text (string).
    - **Description:** Decrypts the input **ciphertext** encrypted using the Caesar cipher with the given **key**. Non-alphabetic characters remain unchanged.

Usage:

1. **Encrypting a Message:**
    - The user is prompted to input a message to be encrypted.
    - The message is encrypted using the Caesar cipher with a fixed key of 11.
    - The encrypted message is printed.
2. **Decrypting the Encrypted Message:**
    - The script automatically decrypts the encrypted message using all possible keys from 1 to 26.
    - The decrypted messages for each key are printed in a decryption table.

Algorithm Description:

Encryption Algorithm (**caesar_encrypt** function):

1. Initialize an empty list **encrypted_text** to store the encrypted characters.
2. For each character **char** in the **plaintext**:

   a. If **char** is an alphabet (checked using **isalpha()**), determine its case (lowercase or uppercase).

   b. Calculate the base ASCII value (**base**) for the case.

   c. Encrypt the character using the formula: **((ASCII(char) - base + key) % 26) + base**.

   d. Append the encrypted character to the **encrypted_text** list.

   e. If **char** is not an alphabet, append it unchanged to the **encrypted_text** list.

3. Join the characters in **encrypted_text** to form the encrypted message and return it.

Decryption Algorithm (**caesar_decrypt** function):

1. Initialize an empty list **decrypted_text** to store the decrypted characters.
2. For each character **char** in the **ciphertext**:

   a. If **char** is an alphabet (checked using **isalpha()**), determine its case (lowercase or uppercase).

   b. Calculate the base ASCII value (**base**) for the case.

   c. Decrypt the character using the formula: **((ASCII(char) - base - key) % 26) + base**.

   d. Append the decrypted character to the **decrypted_text** list.

   e. If **char** is not an alphabet, append it unchanged to the **decrypted_text** list.

3. Join the characters in **decrypted_text** to form the decrypted message and return it.

Example:

```
Write message to encrypt: Kromka

Encrypted Message: Vczxvl

Decryption Table:

Key 1: Ubywuk

Key 2: Taxvtj

Key 3: Szwusi

Key 4: Ryvtrh

Key 5: Qxusqg

Key 6: Pwtrpf

Key 7: Ovsqoe

Key 8: Nurpnd
```

Key 9: Mtqomc

Key 10: Lspnlb

Key 11: Kromka

Key 12: Jqnljz

Key 13: Ipmkiy

Key 14: Holjhx

Key 15: Gnkigw

Key 16: Fmjhfv

Key 17: Eligeu

Key 18: Dkhfdt

Key 19: Cjgecs

Key 20: Bifdbr

Key 21: Ahecaq

Key 22: Zgdbzp

Key 23: Yfcayo

Key 24: Xebzxn

Key 25: Wdaywm

Key 26: Vczxvl