Title: Write a program to implement caesar cipher.

Theory:

- ❖ **Introduction and Definition**

  The Caesar Cipher is a simple and well-known encryption technique that falls under the category of substitution ciphers. It was named after Julius Caesar, who reportedly used it to encrypt military messages. The cipher works by shifting each letter in the plaintext a fixed number of places in the alphabet, making it an example of a monoalphabetic substitution cipher.

- ❖ **Key Concepts and Parameters**
  - Plaintext: The original message that needs to be encrypted.
  - Ciphertext: The encoded message after applying the shift.
  - Shift Key (k): The number of positions each letter in the plaintext is shifted in the alphabet.
  - Encryption: The process of converting plaintext into ciphertext.
  - Decryption: The process of converting ciphertext back into plaintext.

- ❖ **Mathematical Representation**

  The encryption and decryption process of the Caesar Cipher can be mathematically represented as:

  Encryption:

  $$C = (P + K) \bmod 26$$

  Decryption:

  $$P = (C - K) \bmod 26$$

Where:

- P is the numerical representation of a plaintext letter (A = 0, B = 1, ..., Z = 25).
- C is the numerical representation of the corresponding ciphertext letter.
- K is the shift key.
- mod 26 ensures the letters wrap around in the alphabet.

Algorithm:

a. Choose a shift key (e.g., 3).
b. Replace each letter in the plaintext with a letter shifted by the key in the alphabet.
c. Wrap around if the shift exceeds 'Z' or 'z'.
d. Decrypt by shifting in the opposite direction.

Source Code:

//Encryption

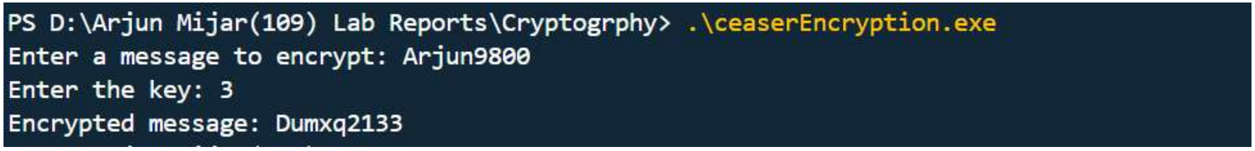
```
#include<stdio.h>
#include<ctype.h>
int main() {
    char text[500], ch;
    int key;

    // Taking user input.
    printf("Enter a message to encrypt: ");
    scanf("%s", text);
    printf("Enter the key: ");
    scanf("%d", & key);

    // Visiting character by character.
    for (int i = 0; text[i] != '\0'; ++i) {
        ch = text[i];
        // Check for valid characters.
        if (isalnum(ch)) {
            //Lowercase characters.
            if (islower(ch)) {
                ch = (ch - 'a' + key) % 26 + 'a';
            }
            // Uppercase characters.
            if (isupper(ch)) {
                ch = (ch - 'A' + key) % 26 + 'A';
            }

            // Numbers.
            if (isdigit(ch)) {
                ch = (ch - '0' + key) % 10 + '0';
            }
        }
        // Invalid character.
        else {
            printf("Invalid Message");
        }

        // Adding encoded answer.
        text[i] = ch;
    }
    printf("Encrypted message: %s", text);

    return 0;
}
```

//Decryption

```
#include<stdio.h>
#include<ctype.h>
int main() {
    char text[500], ch;
    int key;

    // Taking user input.
    printf("Enter a message to encrypt: ");
    scanf("%s", text);
    printf("Enter the key: ");
    scanf("%d", & key);

    // Visiting character by character.

    for (int i = 0; text[i] != '\0'; ++i) {
        ch = text[i];
        // Check for valid characters.
        if (isalnum(ch)) {
            //Lowercase characters.
            if (islower(ch)) {
                ch = (ch - 'a' + key) % 26 + 'a';
            }
            // Uppercase characters.
            if (isupper(ch)) {
                ch = (ch - 'A' + key) % 26 + 'A';
            }


            // Numbers.
            if (isdigit(ch)) {
                ch = (ch - '0' + key) % 10 + '0';
            }
        }
        // Invalid character.
        else {
            printf("Invalid Message");
        }

        // Adding encoded answer.
        text[i] = ch;
    }

    printf("Encrypted message: %s", text);

    return 0;
}
```

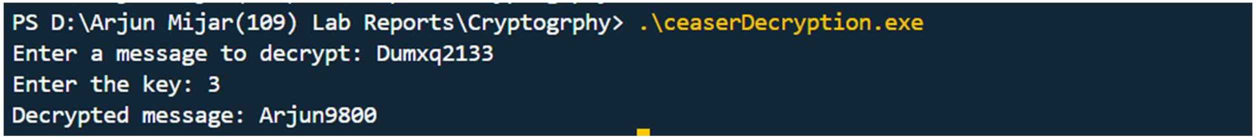
Output:

Encryption:



```
PS D:\Arjun Mijar(109) Lab Reports\Cryptogrphy> .\ceaserEncryption.exe
Enter a message to encrypt: Arjun9800
Enter the key: 3
Encrypted message: Dumxq2133
```

Decryption:



```
PS D:\Arjun Mijar(109) Lab Reports\Cryptogrphy> .\ceaserDecryption.exe
Enter a message to decrypt: Dumxq2133
Enter the key: 3
Decrypted message: Arjun9800
```

Analysis:

The Caesar Cipher is a simple substitution cipher that is easy to implement but vulnerable to brute-force attacks due to its limited number of possible shifts (25). It is primarily used for educational purposes and basic encoding tasks rather than secure encryption.