practical1a.py

```python
import random


class Cryptography:
    def __init__(self, key: int) -> None:
        self.__key: int = key
        self.__plain_text: str = ""
        self.__cypher_text: str = ""

    def encryption(self, input_string: str) -> None:
        self.__cypher_text = ""
        for i in input_string.lower():
            a: int = ord(i) - 97 + self.__key
            index: int = a % 26
            self.__cypher_text += chr(index + 65)
        self.display("Encryption", self.__cypher_text)

    def decryption(self, input_string: str) -> None:
        self.__plain_text = ""
        for i in input_string.upper():
            a: int = ord(i) - 65 - self.__key
            index: int = a % 26
            self.__plain_text += chr(index + 97)
        self.display("Decryption", self.__plain_text)

    @staticmethod
    def display(method: str, text: str) -> None:
        print(f"\n{method} : {text}")


class VigenereCipher:
    def __init__(self) -> None:
        self.__plaintext: str = ""
        self.__cyphertext: str = ""
        self.__keyword: str = ""

    def encryption(self, plaintext: str, keyword: str) -> str:
        self.__plaintext = plaintext.lower()
        self.__keyword = (keyword * (len(self.__plaintext) // len(keyword) + 1))[
            : len(self.__plaintext)
        ]
        self.__cyphertext = ""

        for i, v in enumerate(self.__plaintext):
            if v.isalpha():
                shift = (ord(v) - 97 + ord(self.__keyword[i].lower()) - 97) % 26
                self.__cyphertext += chr(shift + 65)  # Convert to uppercase
            else:
                self.__cyphertext += v

        return self.__cyphertext

    def decryption(self, cyphertext: str, keyword: str) -> str:
        self.__cyphertext = cyphertext.upper()
        self.__keyword = (keyword * (len(self.__cyphertext) // len(keyword) + 1))[
            : len(self.__cyphertext)
        ]
        self.__plaintext = ""

        for i, v in enumerate(self.__cyphertext):
            if v.isalpha():
                shift = (ord(v) - 65 - (ord(self.__keyword[i].lower()) - 97)) % 26
                self.__plaintext += chr(shift + 97)  # Convert to lowercase
            else:
                self.__plaintext += v

        return self.__plaintext


def main() -> None:
    while True:
        choice: int = int(
            input(
                """
    1. Caesar Cipher
    2. Modified Caesar Cipher
    3. Vigenère Cipher
    0. Exit
    Enter Choice: """
            )
        )

        if choice == 0:
            break
        elif choice == 1:
            ceaser_cipher: Cryptography = Cryptography(3)
            input_string: str = input("Enter Your String: ")
            if input_string.islower():
                ceaser_cipher.encryption(input_string)
            else:
                ceaser_cipher.decryption(input_string)
        elif choice == 2:
            key: int = int(input("Enter Your Key: "))
            modified_ceaser_cipher: Cryptography = Cryptography(key)
            input_string: str = input("Enter Your String: ")
            if input_string.islower():
                modified_ceaser_cipher.encryption(input_string)
            else:
                modified_ceaser_cipher.decryption(input_string)
        elif choice == 3:
            keyword: str = input("Enter Your Keyword: ")
            vigenere_cipher: VigenereCipher = VigenereCipher()
            input_string: str = input("Enter Your String: ")
            if input_string.islower():
                encrypted: str = vigenere_cipher.encryption(input_string, keyword)
                print("Encrypted string:", encrypted)
            else:
                decrypted: str = vigenere_cipher.decryption(input_string, keyword)
                print("Decrypted string:", decrypted)
        else:
            print("Invalid Choice")


if __name__ == "__main__":
    main()
```