

practical1a.py

```
1 import random
2
3
4 class Cryptography:
5     def __init__(self, key: int) → None:
6         self.__key: int = key
7         self.__plain_text: str = ""
8         self.__cypher_text: str = ""
9
10    def encryption(self, input_string: str) → None:
11        self.__cypher_text = ""
12        for i in input_string.lower():
13            a: int = ord(i) - 97 + self.__key
14            index: int = a % 26
15            self.__cypher_text += chr(index + 65)
16        self.display("Encryption", self.__cypher_text)
17
18    def decryption(self, input_string: str) → None:
19        self.__plain_text = ""
20        for i in input_string.upper():
21            a: int = ord(i) - 65 - self.__key
22            index: int = a % 26
23            self.__plain_text += chr(index + 97)
24        self.display("Decryption", self.__plain_text)
25
26    @staticmethod
27    def display(method: str, text: str) → None:
28        print(f"\n{method} : {text}")
29
30
31 class VigenereCipher:
32     def __init__(self) → None:
33         self.__plaintext: str = ""
34         self.__cyphertext: str = ""
35         self.__keyword: str = ""
36
37    def encryption(self, plaintext: str, keyword: str) → str:
38        self.__plaintext = plaintext.lower()
39        self.__keyword = (keyword * (len(self.__plaintext) // len(keyword) + 1))[: len(self.__plaintext)]
40        ]
41        self.__cyphertext = ""
42
43
44    for i, v in enumerate(self.__plaintext):
45        if v.isalpha():
46            shift = (ord(v) - 97 + ord(self.__keyword[i].lower()) - 97) % 26
47            self.__cyphertext += chr(shift + 65) # Convert to uppercase
48        else:
```

```

49         self.__cyphertext += v
50
51     return self.__cyphertext
52
53 def decryption(self, cyphertext: str, keyword: str) → str:
54     self.__cyphertext = cyphertext.upper()
55     self.__keyword = (keyword * (len(self.__cyphertext) // len(keyword) + 1))[
56         : len(self.__cyphertext)
57     ]
58     self.__plaintext = ""
59
60     for i, v in enumerate(self.__cyphertext):
61         if v.isalpha():
62             shift = (ord(v) - 65 - (ord(self.__keyword[i].lower()) - 97)) % 26
63             self.__plaintext += chr(shift + 97) # Convert to lowercase
64         else:
65             self.__plaintext += v
66
67     return self.__plaintext
68
69
70 def main() → None:
71     while True:
72         choice: int = int(
73             input(
74                 """
75 1. Caesar Cipher
76 2. Modified Caesar Cipher
77 3. Vigenère Cipher
78 0. Exit
79 Enter Choice: """
80             )
81         )
82
83         if choice == 0:
84             break
85         elif choice == 1:
86             ceaser_cipher: Cryptography = Cryptography(3)
87             input_string: str = input("Enter Your String: ")
88             if input_string.islower():
89                 ceaser_cipher.encryption(input_string)
90             else:
91                 ceaser_cipher.decryption(input_string)
92         elif choice == 2:
93             key: int = int(input("Enter Your Key: "))
94             modified_ceaser_cipher: Cryptography = Cryptography(key)
95             input_string: str = input("Enter Your String: ")
96             if input_string.islower():
97                 modified_ceaser_cipher.encryption(input_string)
98             else:

```

```
99         modified_ceaser_cipher.decryption(input_string)
100     elif choice == 3:
101         keyword: str = input("Enter Your Keyword: ")
102         vigenere_cipher: VigenereCipher = VigenereCipher()
103         input_string: str = input("Enter Your String: ")
104         if input_string.islower():
105             encrypted: str = vigenere_cipher.encryption(input_string, keyword)
106             print("Encrypted string:", encrypted)
107         else:
108             decrypted: str = vigenere_cipher.decryption(input_string, keyword)
109             print("Decrypted string:", decrypted)
110     else:
111         print("Invalid Choice")
112
113
114 if __name__ == "__main__":
115     main()
116
```