Implementation of Classical Ciphers in Python

Task **1**

```
def encrypt(text, shift):
    cipherText = ""

    for char in text:
        if char.isalpha():
            if char.isupper():  # To check the character is upper or lower
                base = 'A'
            else:
                base = 'a'

            char_number = ord(char) - ord(base) # We taking the ASCII accoding to the base
            shifted_number = char_number + shift  # shifting the Index according to the key
            new_position = shifted_number % 26
            new_char = chr(new_position + ord(base)) # Now, we take the ASCII of the new index
            cipherText += new_char # concate the letter of new index to form a cipher text

        else:
              cipherText += char

    return cipherText

def decrypt(text, shift):
    plainText = ""

    for char in text:
        if char.isalpha():
            if char.isupper():  # To check the character is upper or lower
                base = 'A'
            else:
                base = 'a'

            char_number = ord(char) - ord(base)
            shifted_number = char_number - shift  # here, we are substracting the key from the ASCII inorder to get the PlainText
            new_position = shifted_number % 26
            new_char = chr(new_position + ord(base))
            plainText += new_char # concate the letter of new index to form a Plain text
        else:
            plainText += char

    return plainText

text = input("Enter the text to be encypted : ")
key = int(input("Enter the shift value : "))

cipherText = encrypt(text, key)
print("Cipher Text:", cipherText)

plainText = decrypt(cipherText, key)
print("Plain Text:", plainText)
```

```
Enter the text to be encypted : Curating a Responsible Digital World 2025
Enter the shift value : 5
Cipher Text: Hzwfynsl f Wjxutsxngqj Inlnyfq Btwqi 2025
Plain Text: Curating a Responsible Digital World 2025
```

Task 2

```
def encrypt(text, key): #function to encrypt
    cipherText = ""

    for i in range(len(text)): # iterate through the string
        char = text[i]
        if char.isalpha():
            base = 'A' if char.isupper() else 'a' #to check whether the character
            key_shift = ord(key[i]) - ord(base) # taking the ASCII value of each character in key and base, substracting them and storir
            new_alpha = (ord(char) - ord(base) + key_shift) # adding the plaintext and key
            new_char = chr(new_alpha % 26 + ord(base)) # equation of vignere cipher - (plaintext + key number) % 26
            cipherText += new_char
        else:
            cipherText += char
```

```
    return cipherText

def decrypt(text, key):
    plainText = ""

    for i in range(len(text)):
        char = text[i]
        if char.isalpha():
            base = 'A' if char.isupper() else 'a'
            key_shift = ord(key[i]) - ord(base)
            new_alpha = (ord(char) - ord(base) - key_shift)
            new_char = chr(new_alpha % 26 + ord(base))
            plainText += new_char
        else:
            plainText += char

    return plainText

def extend_key(text, key):
    extended_keyword = ""
    i = 0

    for char in text:
        extended_keyword += key[i]
        i += 1
        if i == len(key):
            i = 0

    return extended_keyword

plainText = input("Enter the text: ")
key = input("Enter the key: ")

extended_key = extend_key(plainText, key)
print("Extended Key:", extended_key)

cipherText = encrypt(plainText, extended_key)
print("Cipher Text:", cipherText)

decryptedText = decrypt(cipherText, extended_key)
print("Decrypted Text:", decryptedText)
```

```
Enter the text: curating a responsible digital world
Enter the key: key
Extended Key: keykeykeykeykeykeykeykeykeykeykey
Cipher Text: mypkxgxk k pownyrqsfjo bskgdej ambpb
Decrypted Text: curating a responsible digital world
```

Task - 3

```
import random

def newKey(size):
    key = ""
    for _ in range(size):
        random_letter = chr(random.randint(65, 90))  # Random letter from 'A' to 'Z'
        key += random_letter #appending the new character to form a word (key)
    return key


def encrypt(text, key): #function to encrypt
    cipherText = ""

    for i in range(len(text)): # iterate through the string
        char = text[i]
        if char.isalpha():
            base = 'A' if char.isupper() else 'a' #to check whether the character
            key_shift = ord(key[i]) - ord(base) # taking the ASCII value of each character in key and base, substracting them and storing
            new_alpha = (ord(char) - ord(base) + key_shift) # adding the plaintext and key
            new_char = chr(new_alpha % 26 + ord(base)) # equation of vignere cipher - (plaintext + key number) % 26
            cipherText += new_char
        else:
            cipherText += char

    return cipherText

def decrypt(text, key):
    plainText = ""
```

```
    for i in range(len(text)):
        char = text[i]
        if char.isalpha():
            base = 'A' if char.isupper() else 'a'
            key_shift = ord(key[i]) - ord(base)
            new_alpha = (ord(char) - ord(base) - key_shift)
            new_char = chr(new_alpha % 26 + ord(base))
            plainText += new_char
        else:
            plainText += char

    return plainText


plainText = input("Enter the text: ")
key = int(input("Enter the key size that of the plain text: "))

n = len(plainText)

if key < n:
  print("Change the key size!! ")
  exit()

else:
  extended_key = newKey(key)

  cipherText = encrypt(plainText, extended_key)
  print("Cipher Text:", cipherText)

  decryptedText = decrypt(cipherText, extended_key)
  print("Decrypted Text:", decryptedText)
```

```
Enter the text: nkv
Enter the key size that of the plain text: 3
Cipher Text: pfg
Decrypted Text: nkv
```