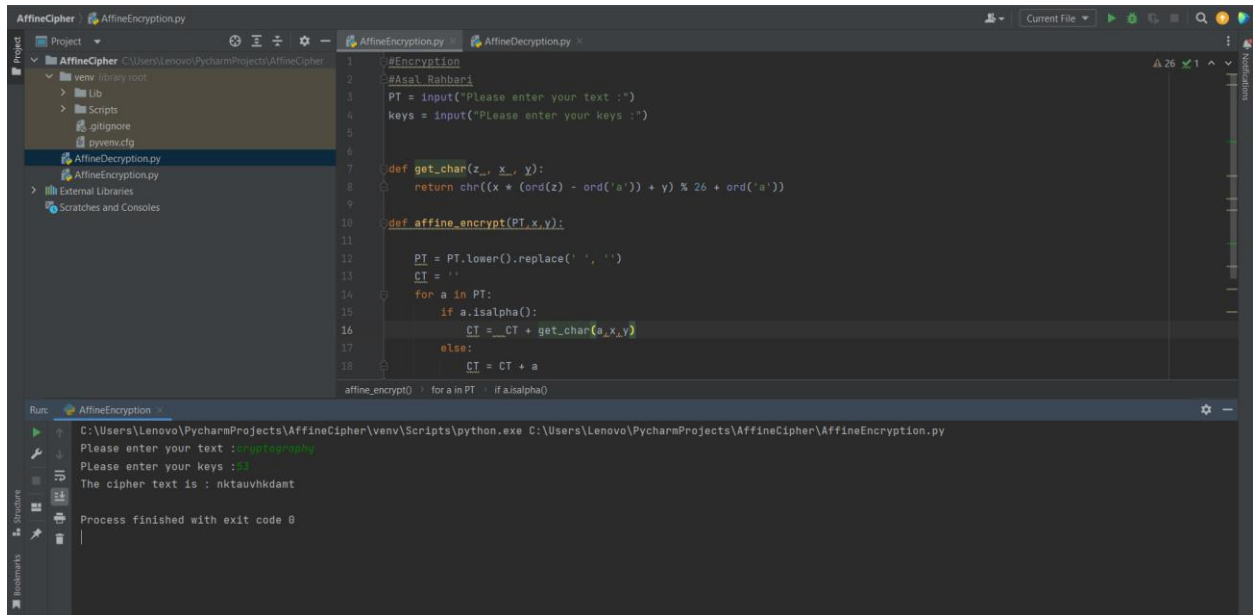


# Affine Cipher – Asal Rahbari

Encryption :

Plaintext : cryptography

Keys : 5 , 3

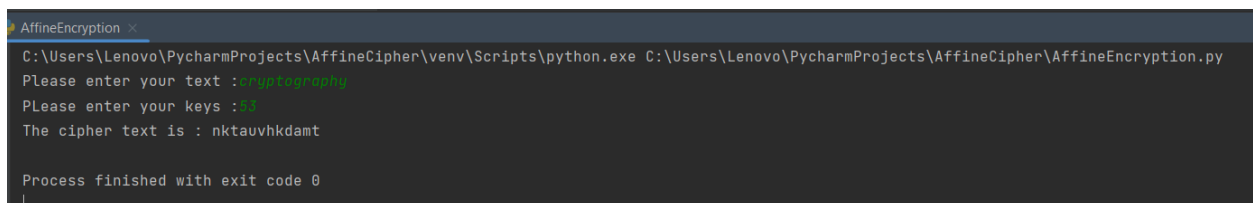


The screenshot shows the PyCharm IDE with the project 'AffineCipher' open. The file 'AffineEncryption.py' is selected in the Project tool window. The code in the editor is as follows:

```
1 #Encryption
2 #Asal_Rahbari
3 PT = input("Please enter your text :")
4 keys = input("Please enter your keys :")
5
6
7 def get_char(z, x, y):
8     return chr((x * (ord(z) - ord('a')) + y) % 26 + ord('a'))
9
10 def affine_encrypt(PT, x, y):
11
12     PT = PT.lower().replace(' ', '')
13     CT = ''
14     for a in PT:
15         if a.isalpha():
16             CT = CT + get_char(a, x, y)
17         else:
18             CT = CT + a
19
20 affine_encrypt(PT, x, y)
```

The Run tool window shows the execution of the script:

```
Run: AffineEncryption
C:\Users\Lenovo\PycharmProjects\AffineCipher\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\AffineCipher\AffineEncryption.py
Please enter your text :cryptography
Please enter your keys :53
The cipher text is : nktauvhkdamt
Process finished with exit code 0
```



The screenshot shows a terminal window titled 'AffineEncryption' with the following output:

```
C:\Users\Lenovo\PycharmProjects\AffineCipher\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\AffineCipher\AffineEncryption.py
Please enter your text :cryptography
Please enter your keys :53
The cipher text is : nktauvhkdamt
Process finished with exit code 0
```

Decryption :

Ciphertext : nktauvhkdamt

Keys : 5 , 3

The screenshot shows the PyCharm IDE with the `AffineDecryption.py` file open. The code implements a decryption function for an affine cipher. The `gcd` function is defined to find the greatest common divisor of two numbers. The `multiplicative_inverse` function is defined to calculate the multiplicative inverse of a number modulo 26. The main execution flow is as follows:

```
1 #Decryption
2 #Asal_Rahbari
3 CT = input("Please enter your text :)")
4 keys = input("Please enter your keys :)")
5 #find the gcd
6 def gcd(a, b):
7     if a == 0:
8         return (b, 0, 1)
9     else:
10        g, y, x = gcd(b % a, a)
11        return (g, x - (b // a) * y, y)
12 #define a function to calculate the multiplicative inverse of the first given key in the mod 26
13
14 def multiplicative_inverse(x, y):
15     a, z, t = gcd(x, y)
16     if a != 1:
17         raise Exception("The given number doesn't have inverse in mod 26 ")
18     else:
```

The Run console shows the execution of the script:

```
Run: AffineDecryption.py
C:\Users\Lenovo\PycharmProjects\AffineCipher\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\AffineCipher\AffineDecryption.py
Please enter your text :nktauvhkdamt
Please enter your keys :11
The plaintext is : cryptography
Process finished with exit code 0
```

The terminal window shows the execution of the script:

```
AffineDecryption x
C:\Users\Lenovo\PycharmProjects\AffineCipher\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\AffineCipher\AffineDecryption.py
Please enter your text :nktauvhkdamt
Please enter your keys :11
The plaintext is : cryptography
Process finished with exit code 0
```

Brute-Force Attack :

Cipher text : nktauvhkdamt

```

AffineCipher AffineAttack.py
Project
Run: AffineAttack AffineEncryption
C:\Users\Lenovo\PycharmProjects\AffineCipher\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\AffineCipher\AffineAttack.py
Please enter your text :nktavvhkdamt
a = 1, b = 0: nktavvhkdamt
a = 1, b = 1: mjsztugjczls
a = 1, b = 2: lirystfbykr
a = 1, b = 3: khqxrsehaxjq
a = 1, b = 4: jgpwgrdgzwp
a = 1, b = 5: ifovpqcryvho
a = 1, b = 6: henuopbexugn
a = 1, b = 7: gdmtnoadwtfm
a = 1, b = 8: fclsmnzcvsel
a = 1, b = 9: ebkrlybburdk
a = 1, b = 10: dajqkLxatqej
a = 1, b = 11: czipjkwzspbi
a = 1, b = 12: byhoijvyroah
a = 1, b = 13: axgnhluxqzng
a = 1, b = 14: zwfeghtwpmayf
a = 1, b = 15: yvelfgsvolxe
a = 1, b = 16: xudkefrunkwd
a = 1, b = 17: wtcjdeqtmjvc
a = 1, b = 18: vsbicdpsliub
a = 1, b = 19: urahbcorkhta
a = 1, b = 20: tqzgabnqjgsz
a = 1, b = 21: spyfzampifry
a = 1, b = 22: roxezyloheqx
a = 1, b = 23: qmxdxykngdpw
a = 1, b = 24: pmvcwxjmfcoy
a = 1, b = 25: olubvialebnu
a = 3, b = 0: nmpayhlmbaep
a = 3, b = 1: edgrppycdsrvq

```

a = 5, b = 3: cryptography