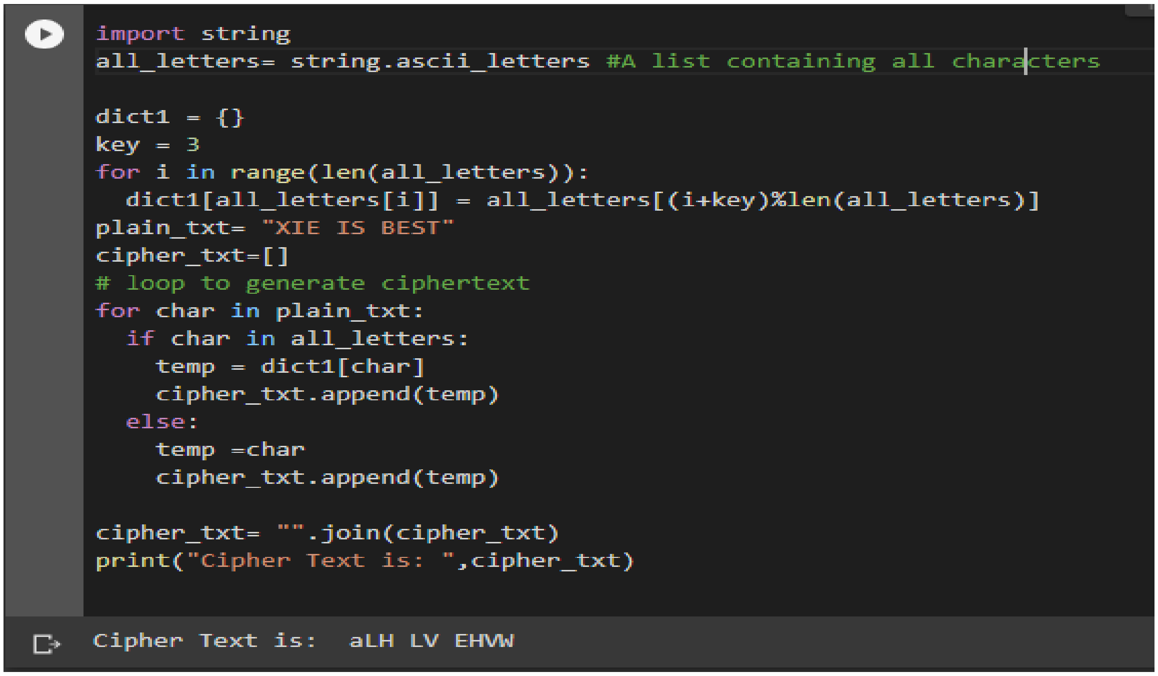
1. Caesar Cipher Encryption & Decryption

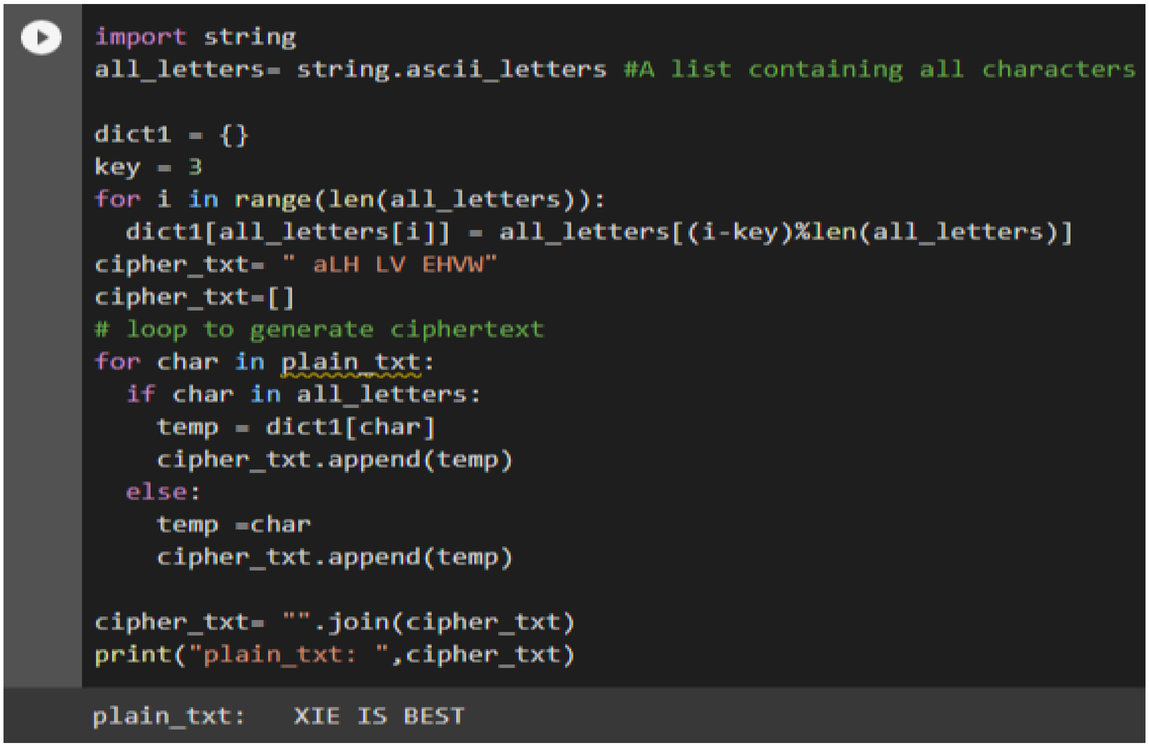


```
import string
all_letters= string.ascii_letters #A list containing all characters

dict1 = {}
key = 3
for i in range(len(all_letters)):
  dict1[all_letters[i]] = all_letters[(i+key)%len(all_letters)]
plain_txt= "XIE IS BEST"
cipher_txt=[]
# loop to generate ciphertext
for char in plain_txt:
  if char in all_letters:
    temp = dict1[char]
    cipher_txt.append(temp)
  else:
    temp =char
    cipher_txt.append(temp)

cipher_txt= "".join(cipher_txt)
print("Cipher Text is: ",cipher_txt)
```

```
Cipher Text is:  aLH LV EHVW
```



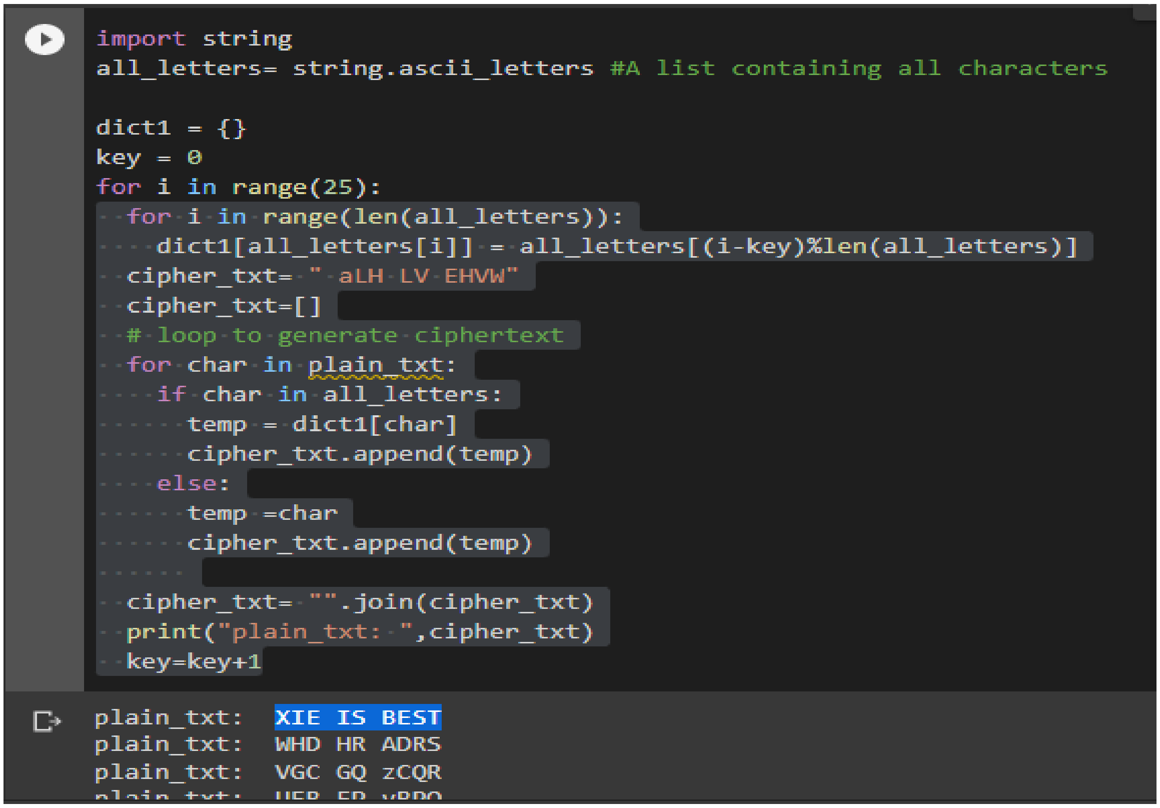
```
import string
all_letters= string.ascii_letters #A list containing all characters

dict1 = {}
key = 3
for i in range(len(all_letters)):
  dict1[all_letters[i]] = all_letters[(i-key)%len(all_letters)]
cipher_txt= " aLH LV EHVW"
cipher_txt=[]
# loop to generate ciphertext
for char in plain_txt:
  if char in all_letters:
    temp = dict1[char]
    cipher_txt.append(temp)
  else:
    temp =char
    cipher_txt.append(temp)

cipher_txt= "".join(cipher_txt)
print("plain_txt: ",cipher_txt)
```

```
plain_txt:   XIE IS BEST
```

2. Caesar Cipher Encryption & Decryption

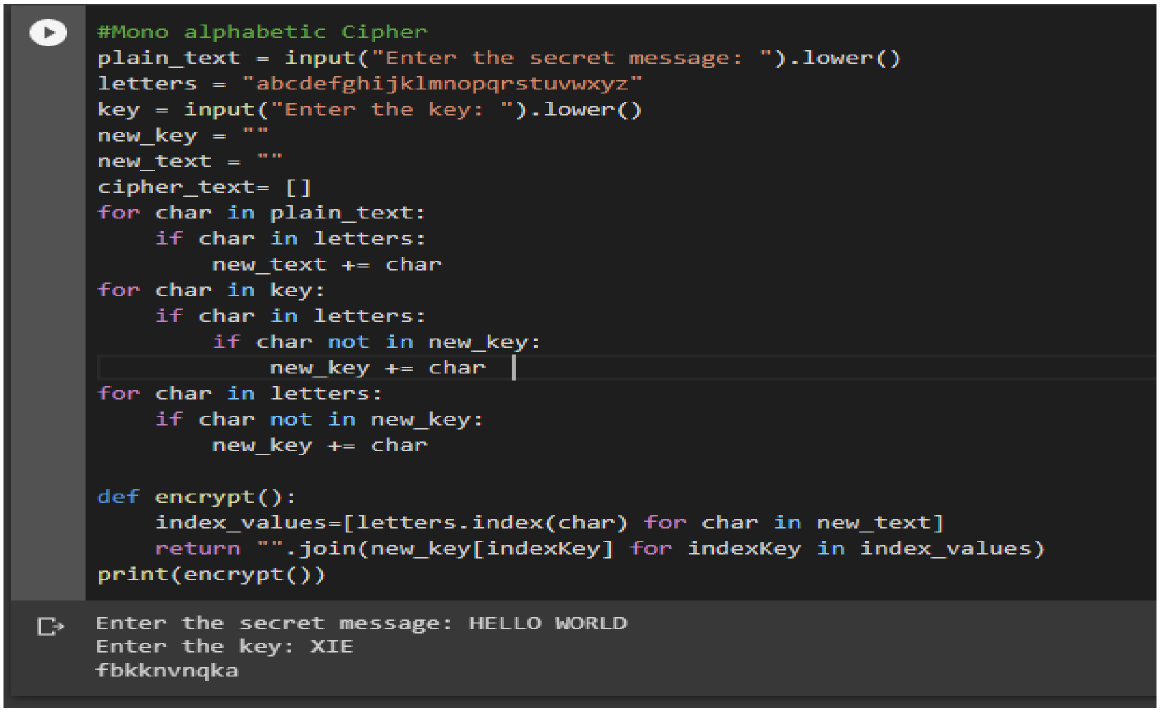


```
import string
all_letters= string.ascii_letters #A list containing all characters

dict1 = {}
key = 0
for i in range(25):
  for i in range(len(all_letters)):
    dict1[all_letters[i]] = all_letters[(i-key)%len(all_letters)]
  cipher_txt= " aLH LV EHVW"
  cipher_txt=[]
  # loop to generate ciphertext
  for char in plain_txt:
    if char in all_letters:
      temp = dict1[char]
      cipher_txt.append(temp)
    else:
      temp =char
      cipher_txt.append(temp)
      
  cipher_txt= "".join(cipher_txt)
  print("plain_txt: ",cipher_txt)
  key=key+1
```

```
plain_txt:  XIE IS BEST
plain_txt:  WHD HR ADRS
plain_txt:  VGC GQ zCQR
plain_txt:  UFB FP yBPQ
```

3. Mono Alphabetic Cipher

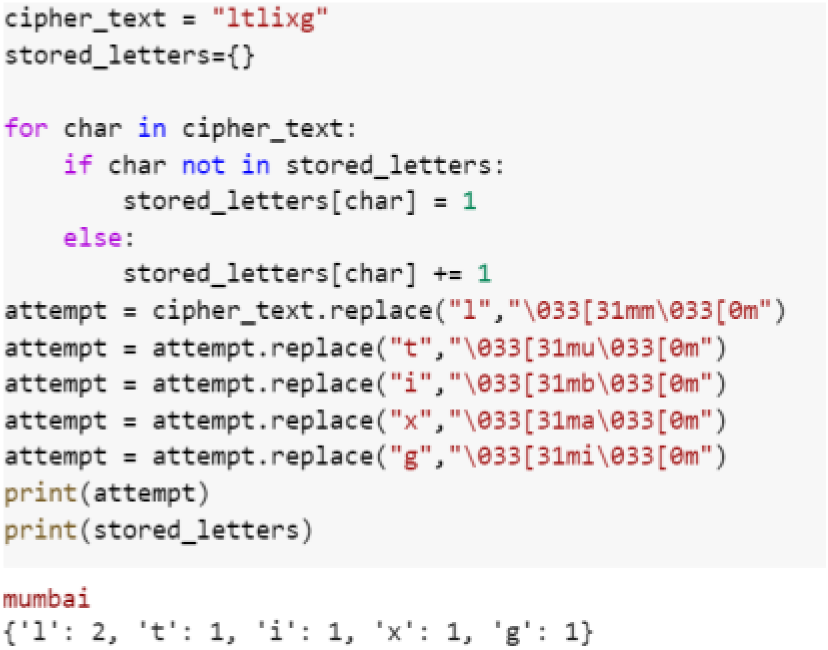


```
#Mono alphabetic Cipher
plain_text = input("Enter the secret message: ").lower()
letters = "abcdefghijklmnopqrstuvwxyz"
key = input("Enter the key: ").lower()
new_key = ""
new_text = ""
cipher_text= []
for char in plain_text:
    if char in letters:
        new_text += char
for char in key:
    if char in letters:
        if char not in new_key:
            new_key += char
for char in letters:
    if char not in new_key:
        new_key += char

def encrypt():
    index_values=[letters.index(char) for char in new_text]
    return "".join(new_key[indexKey] for indexKey in index_values)
print(encrypt())
```

```
Enter the secret message: HELLO WORLD
Enter the key: XIE
fbkknvnqka
```

4. Guessing Attack using Frequency Analysis on Mono Alphabetic Cipher.



```
cipher_text = "ltlixg"
stored_letters={}

for char in cipher_text:
    if char not in stored_letters:
        stored_letters[char] = 1
    else:
        stored_letters[char] += 1
attempt = cipher_text.replace("l","\033[31mm\033[0m")
attempt = attempt.replace("t","\033[31mu\033[0m")
attempt = attempt.replace("i","\033[31mb\033[0m")
attempt = attempt.replace("x","\033[31ma\033[0m")
attempt = attempt.replace("g","\033[31mi\033[0m")
print(attempt)
print(stored_letters)
```

```
mumbai
{'l': 2, 't': 1, 'i': 1, 'x': 1, 'g': 1}
```