```python
def decrypt():
    encrypted_message = input("Enter the message i.e to be decrypted
        : ").strip()
    lowercase_letters = "abcdefghijklmnopqrstuvwxyz"
    uppercase_letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    k = int(input("Enter the key to decrypt: "))
    decrypted_message = ""

    for ch in encrypted_message:
        if ch in lowercase_letters:
            position = lowercase_letters.find(ch)
            new_pos = (position - k) % 26
            new_char = lowercase_letters[new_pos]
            decrypted_message += new_char
        elif ch in uppercase_letters:
            position = uppercase_letters.find(ch)
            new_pos = (position - k) % 26
            new_char = uppercase_letters[new_pos]
            decrypted_message += new_char
        else:
            decrypted_message += ch

    print("Your decrypted message is:\n")
    print(decrypted_message)

decrypt()
```

```
Enter the message i.e to be decrypted: PHHW
Enter the key to decrypt: 3
Your decrypted message is:

MEET

=== Code Execution Successful ===
```

```python
message = 'RD SFRJ NX WFLMZ'
Letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
for key in range(len(Letters)):
   translated = ''
   for ch in message:
      if ch in Letters:
         num = Letters.find(ch)
         num = num - key
         if num < 0:
            num = num + len(Letters)
         translated = translated + Letters[num]
      else:
         translated = translated + ch
   print('Hacking key is %s: %s' % (key, translated))
```

```
Hacking key is 0: RD SFRJ NX WFLMZ
Hacking key is 1: QC REQI MW VEKLY
Hacking key is 2: PB QDPH LV UDJKX
Hacking key is 3: OA PCOG KU TCIJW
Hacking key is 4: NZ OBNF JT SBHIV
Hacking key is 5: MY NAME IS RAGHU
Hacking key is 6: LX MZLD HR QZFGT
Hacking key is 7: KW LYKC GQ PYEFS
Hacking key is 8: JV KXJB FP OXDER
Hacking key is 9: IU JWIA EO NWCDQ
Hacking key is 10: HT IVHZ DN MVBCP
Hacking key is 11: GS HUGY CM LUABO
Hacking key is 12: FR GTFX BL KTZAN
Hacking key is 13: EQ FSEW AK JSYZM
Hacking key is 14: DP ERDV ZJ IRXYL
Hacking key is 15: CO DQCU YI HQWXK
Hacking key is 16: BN CPBT XH GPVWJ
Hacking key is 17: AM BOAS WG FOUVI
Hacking key is 18: ZL ANZR VF ENTUH
Hacking key is 19: YK ZMYQ UE DMSTG
Hacking key is 20: XJ YLXP TD CLRSF
Hacking key is 21: WI XKWO SC BKQRE
Hacking key is 22: VH WJVN RB AJPQD
Hacking key is 23: UG VIUM QA ZIOPC
Hacking key is 24: TF UHTL PZ YHNOB
Hacking key is 25: SE TGSK OY XGMNA
```

```python
q = 23
x = 9
print('The prime number is : ',q)
print('The primitive root of q is : ',x)
a = 4
print('The Private Key a for Ram is : ',a)
b = 3
print('The Private Key b for Preethi is : ',b)
s = int(pow(x,a,q))
t = int(pow(x,b,q))
ka = int(pow(t,a,q))
kb = int(pow(s,b,q))
print('Secret key for the Ram is : ',ka)
print('Secret Key for the Preethi is : ',kb)
```

```
The prime number is :  23
The primitive root of q is :  9
The Private Key a for Ram is :  4
The Private Key b for Preethi is :  3
Secret key for the Ram is :  9
Secret Key for the Preethi is :  9

=== Code Execution Successful ===
```

```python
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)
def is_coprime(a, b):
    return gcd(a, b) == 1
def is_valid_affine(a, b):
    return is_coprime(a, 26) and b >= 0 and b < 26
def encrypt_affine(msg, a, b):
    ciphertext = ''
    for c in msg:
        if c.isalpha():
            idx = ord(c.upper()) - ord('A')
            idx = (a * idx + b) % 26
            ciphertext += chr(idx + ord('A'))
        else:
            ciphertext += c
    return ciphertext
msg =input("Enter the plain text: ")
a = 5
b = 7
if is_valid_affine(a, b):
    ciphertext = encrypt_affine(msg, a, b)
    print("plaintext:", msg)
    print("Ciphertext:", ciphertext)
```

```
Enter the plain text: meet
plaintext: meet
Ciphertext: PBBY

=== Code Execution Successful ==
```

```python
P="hello everyone"
lst1= []
plaintext= []
for i in range(97,123):
    lst1.append(chr(i))
print(lst1)
k=[]
for j in lst1:
    k.append(lst1.index(j))
print(k)
for i in P:
    if i in lst1:
        print(lst1.index(i))
        plaintext.append(lst1.index(i))
print(plaintext)
cipher=[x+1 for x in plaintext]
print(cipher)
for m in cipher:
    if m in k:
        print(lst1[m],end="")
```

STDIN

Input for the program (Optional)

Output:

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
7
4
11
11
14
4
21
4
17
24
14
13
4
[7, 4, 11, 11, 14, 4, 21, 4, 17, 24, 14, 13, 4]
[8, 5, 12, 12, 15, 5, 22, 5, 18, 25, 15, 14, 5]
ifmmpfwfszpof
```