In [ ]:

*# Implementation of RSA algorithm.*

In [1]:

**def** isPrime(n):  
 **if** n==1:  
 **return** **False**  
 **elif** (n == 2):  
 **return** **True**  
 **else**:  
 **for** i **in** range(2,n):  
 **if** n%**i** == 0:  
 **return** **False**  
 **return** **True**

In [2]:

**def** find\_d(e: int, z: int):  
 *# d -> ed = 1(mod z) ; 1 < d < z*  
 d = 2  
 **while** d < z:  
 **if** ((d\*e) % z) == 1:  
 **return** d  
 **else**:  
 d += 1

In [3]:

*# Enter p: 19*  
*# Enter q: 29*  
*# Enter e: 5*  
*# Enter m: 10*

In [4]:

**import** **math**

In [5]:

*# p,q,e,m*  
  
p = int(input("Enter p:"))  
q = int(input("Enter q:"))  
e = int(input("Enter e:"))  
m = int(input("Enter m:"))  
  
**if** isPrime(p) **and** isPrime(q):  
 n = p \* q  
 phi = (p - 1) \* (q - 1)  
 **while** e < phi:  
 **if** math.gcd(e, phi) == 1:  
 **break**  
 **else**:  
 e += 1  
  
d = find\_d(e, phi)  
  
Encrypted = pow(m, e, n)  
Decrypted = pow(Encrypted, d, n)  
  
print("Encrypted : ", Encrypted)  
print("Decrypted : ", Decrypted)

Enter p:19  
Enter q:29  
Enter e:5  
Enter m:10  
Encrypted : 269  
Decrypted : 10

In [5]:

%%**shell**  
jupyter nbconvert --to html /PATH/TO/YOUR/NOTEBOOKFILE.ipynb

In [6]:

**from** **google.colab** **import** drive  
drive.mount('/content/drive')

Mounted at /content/drive