Deffie hellman

```
prime_no=int(input("enter prime no P: "))
prime_no=int(input("enter prime no Q: "))
g=int(input("enter primitive root(g<p): "))
pkxa=int(input("enter private key of A(xa<p): "))
pkxb=int(input("enter private key of B(xb<p): "))
ya=g**pkxa%prime_no
yb=g**pkxb%prime_no
ka=yb**pkxa%prime_no
kb=yb**pkxb%prime_no
print("enter public key ya= ",ya)
print("enter public key yb= ",yb)
print("shared secret key k= ",ka)
```

RSA

```
import math
p=3
q=7
n=p*q
print("n=",n)
phi=(p-1)*(q-1)
e=2
while(e<phi):
    if(math.gcd(e,phi)==1):
        break
    else:
        e+=1
        print("e=",e)
        k=2
        d=((k*phi)+1)/e
        print(f'public key:{e,n}')
        print(f'public key:{d,n}')
        msg=11
        print(f'original message:{msg}')
        c=pow(msg,e)
        c=math.fmod(c,n)
        print(f'Encrypted message:{c}')
        m=pow(c,d)
        m=math.fmod(m,n)
        print(f'Decrypted massage:{m}')
```

AES

```
def meachine():
    keys = 'abcdefghijklmnopqrstuvwxyz |'
    values = keys[-1] + keys[0:-1]
    encryptDict = dict(zip(keys, values))
    dencryptDict = dict(zip(values , keys))
    message = input("enter your secret msg:")
    mode = input("Crypto mode: Encode(E) OR Decode(D)")
    if mode.upper() == 'E':
        newMessage = ''.join([encryptDict[letter]
                              for letter in message.lower()])
    elif mode.upper() == 'D':
        newMessage = ''.join([encryptDict[letter]
                              for letter in message.lower()])
    else:
        print("plz try again...")
    return newMessage.capitalize()
print(meachine())
```