RSA ALGORITHM CODE:

```
def gcd(a, b): # calculates GCD of a and d
    while b != 0:
        c = a % b
        a = b
        b = c
    return a

def modinv(a, m): # calculates modulo inverse of a for mod m
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None

def coprimes(a): # calculates all possible co-prime numbers with a
    l = []
    for x in range(2, a):
        if gcd(a, x) == 1 and modinv(x, phi) != None:
            l.append(x)
    for x in l:
        if x == modinv(x, phi):
            l.remove(x)
    return l
def encrypt_block(m): # encrypts a single block
    c = m ** e % n
    return c

def decrypt_block(c): # decrypts a single block
    m = c ** d % n
    return m

def encrypt_string(s): # applies encryption
    return ''.join([chr(encrypt_block(ord(x))) for x in list(s)])

def decrypt_string(s): # applies decryption
    return ''.join([chr(decrypt_block(ord(x))) for x in list(s)])
if __name__ == "__main__":
```

```
    p = int(input('Enter prime p: '))
    q = int(input('Enter prime q: '))
    print("Choosen primes:\np=" + str(p) + ", q=" + str(q) + "\n")
    n = p * q
    print("n = p * q = " + str(n) + "\n")
    phi = (p - 1) * (q - 1)
    print("Euler's function (totient) [phi(n)]: " + str(phi) + "\n")
    print("Choose an e from a below coprimes array:\n")
    print(str(coprimes(phi)) + "\n")
    e = int(input())
    d = modinv(e, phi) # calculates the decryption key d
    print("\nYour public key is a pair of numbers (e=" + str(e) + ", n=" + str(n) +
").\n")
    print("Your private key is a pair of numbers (d=" + str(d) + ", n=" + str(n) +
").\n")
    s = input("Enter a message to encrypt: ")
    print("\nPlain message: " + s + "\n")
    enc = encrypt_string(s)
    print("Encrypted message: ", enc, "\n")
    dec = decrypt_string(enc)
    print("Decrypted message: " + dec + "\n")
```

OUTPUT:

```
Enter prime p: 11
Enter prime q: 13
Choosen primes:
p=11, q=13

n = p * q = 143

Euler's function (totient) [phi(n)]: 120

Choose an e from a below coprimes array:

[7, 13, 17, 23, 31, 37, 43, 47, 53, 61, 67, 73, 77, 83, 91, 97, 103, 107, 113]

7

Your public key is a pair of numbers (e=7, n=143).

Your private key is a pair of numbers (d=103, n=143).

Enter a message to encrypt: hello

Plain message: hello

Encrypted message:  [>-

Decrypted message: hello



...Program finished with exit code 0
Press ENTER to exit console.
```