Valor de entrada e listas

```
s = input("Digite algo:")
txt = [ord(x) for x in s]
list_e = []
crypto = []
decrypt = []
```

“s” recebe um valor de entrada.

“txt” é uma lista onde está armazenado todos caracteres que “s” recebeu porem em ASCII.

“list_e” lista onde é armazenado valor de E.

“crypto” lista onde está armazenado texto criptografado.

“decrypt” lista onde está armazenado texto descriptografado.

Definição chaves privadas P e Q

```
# P e Q chaves privadas escolhidas pelo usuario sendo P e Q numeros primos
P = 17
Q = 41
```

Chaves privadas P e Q são definidas pelo usuário onde P e Q devem ser números primos.

Definição chave privada N

```
# N chave publica gerada atraves das chaves privadas P e Q.
N = P * Q
```

Chave publica N é gerada através da chave privada P e Q onde:

N = P * Q

Definição chave privada phiN

```
# Definiçao phiN
phiN = (P - 1) * (Q - 1)
```

phiN é uma variável utilizada para descobrir o valor da chave privada D que irá ser descoberto daqui a pouco.

phiN = (P-1) * (Q-1)

Definição chave privada E

```
# Chave publica E
for i in range(phiN):
    prim = gcd(i, phiN)
    if (i > 1) and (i < phiN) and (prim == 1):
        list_e.append(i)
E = random.choice(list_e)
```

Para descobrirmos o valor de “E” fizemos um “for” onde para cada número de 0 a “phiN” ele faz verificação usando a função “gcd” que no caso essa função faz um MCD e sempre que o resultado for igual a 1 significa que é um número primo.

O “if” faz verificação se o “i” é maior que 1 e menor que ”phiN”, e “prim” igual a 1 e adiciona esses valores a “list_e” e logo em seguida é atribuído um valor a “E” pegando randomicamente um valor da “list_e”.

Definição chave privada D

```
# Definicao chave privada D, fazendo calculo inverso modular
D = pow(E, phiN-1, phiN)
```

O valor de “D” é realizo por um cálculo modular multiplicativo inverso.

D*E mod phiN == 1

Criptografar Mensagem

```
# Criptografar msg
for i in txt:
    C = (i ** E) % N
    crypto.append(C)
print(crypto)
```

Parte de criptografar a mensagem é feito um “for” onde pega cada valor da lista “txt” que está em ASCII e faz a segunda conta:

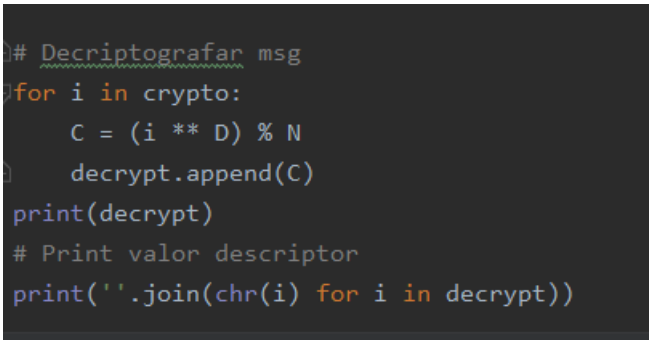
C = (i**E)%N

Basicamente “i” elevado a chave publica “E” e divide por “N” e pega o resto da divisão.

Obs: “%” em Python equivale a mod ou seja resto de uma divisão.

E logo após criptografar os valores são armazenados na lista “crypto”.

Descriptografar Mensagem



```
# Decriptografar msg
for i in crypto:
    C = (i ** D) % N
    decrypt.append(C)
print(decrypt)
# Print valor descriptor
print(''.join(chr(i) for i in decrypt))
```

Para descriptografar a mensagem basicamente acessa os valores de crypto usando um “for” para pegar cada item e faz a seguinte conta:

C = (i**D) % N

Onde “i” é cada mensagem criptografada elevado a “D” que é uma chave privada e mod “N” que basicamente é pegar o resto da divisão.