1)

Message: 2148321
Public Key: (N): 12982131232583, (e): 5
Encoded Message: 6628531511014
Secret Key: 7789272102893
Decoded Message: 2148321
Encoder Working?: True

part iii:
p: 1334737, q: 9726359
part iv:

d: 7789272102893
part v:
Encoded message: 6628531511014
part vi:
Decoded message: 2148321

i)
$$T(n) = 3 T(n/2) + 0$$

$$\begin{cases}
a=3 \\
b=2 \\
d=0
\end{cases}$$

$$d < \log_b a \qquad \therefore T(n) = O(n^{\log_2 3})$$

ii)
$$T(n) = 4 T(n/s) + O(n)$$

iii)
$$T(n) = 4 T(n/2) + O(n^2)$$

, T(o)=0

$$T(n) = T(n-1) + n$$

$$= T(n-2) + n + n$$

$$\vdots$$

$$= n(n)$$

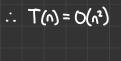
$$= v_5$$
$$= U(u)$$

T(n) = T(n-1) + n

iv)

$$T(n) = O(n^2)$$





Assume its sorted from smallest to largest

def logn(A): 1 = 0r = len(A)-1while $(l \ll r)$: mid = (l+r) // 2if (A[mid] > mid): r = mid - 110 elif (A[mid] < mid):</pre> l = mid + 111 12 else: 13 return True 14 return False

Checks center of the array and eliminates half of the array for analysis if A[i] != i. The center of the remaining half is checked to recursively remove remaining halves until an answer is found.

$$T(n) = T(n/2) + O(1)$$

$$a = 1$$

$$b = 2$$

$$d = \log_b a \implies T(n) = O(n^n \log_b n)$$

$$T(n) = O(\log_b n)$$

$$T(n) = 3T(n/3) + O(n)$$

$$a=3$$

$$b=3$$

$$d=1$$

$$d=1$$

$$C(n \log n)$$

