CRYPTOGRAPHY -BROADCASTS ARE BAD

CODE:

pip install pycryptodome

from Crypto.Util.number import long\_to\_bytes

# Given values

e = 3

ct1 = 78406286729520927933597606356229817207730090348466337528850035686907731769583595879598630905387554797365268804265749503975166670245660278708562880234673953125

n1 = 93500162517048641546787096055793757535647246115573452820070426952722839105259206386726342320373080028589838409420934957286736144380948165832820988099621769638760228520922601557891245459269555695273287461982149159244732888497451549014876606240205619538734235283434466606740831016126239969278127395932813972501

n2 = 69389646088454519961777612972795170975757344514768971340015442739948594078401371455619333135568702919584689527569525412014628819453197654891812622405663009392306477333014587313114349970319506949532748028160803547659404882094671572233644067789871312676667284969540366763124836901936574299355649024020474292507

n3 = 173343492679580453099087472373077831318468581021117934856501363981932357178004338045575880395223727472909876082267313542279706329989182144352883996336740679533245920391761546352887945995038709117785508115723234046039796085997076590595344206748962306765136674210538263394651485897800900761481602355016363501601

# Function to compute cube root using binary search method

def integer\_cube\_root(n):

low = 0

high = n

while low <= high:

mid = (low + high) // 2

if mid\*\*3 == n:

return mid

elif mid\*\*3 < n:

low = mid + 1

else:

high = mid - 1

return None

# Decrypting the ciphertext

def decrypt(ciphertext, modulus):

# Compute cube root of ciphertext

m = integer\_cube\_root(ciphertext)

if m is None:

raise ValueError("Cube root is not an integer.")

# Convert the integer to bytes

plaintext = long\_to\_bytes(m)

return plaintext

# Decrypting ct1 with n1

plaintext1 = decrypt(ct1, n1)

print("Decrypted message from ct1:", plaintext1.decode())

# Decrypting ct2 with n2 (if needed)

plaintext2 = decrypt(ct2, n2)

print("Decrypted message from ct2:", plaintext2.decode())

# Decrypting ct3 with n3 (if needed)

plaintext3 = decrypt(ct3, n3)

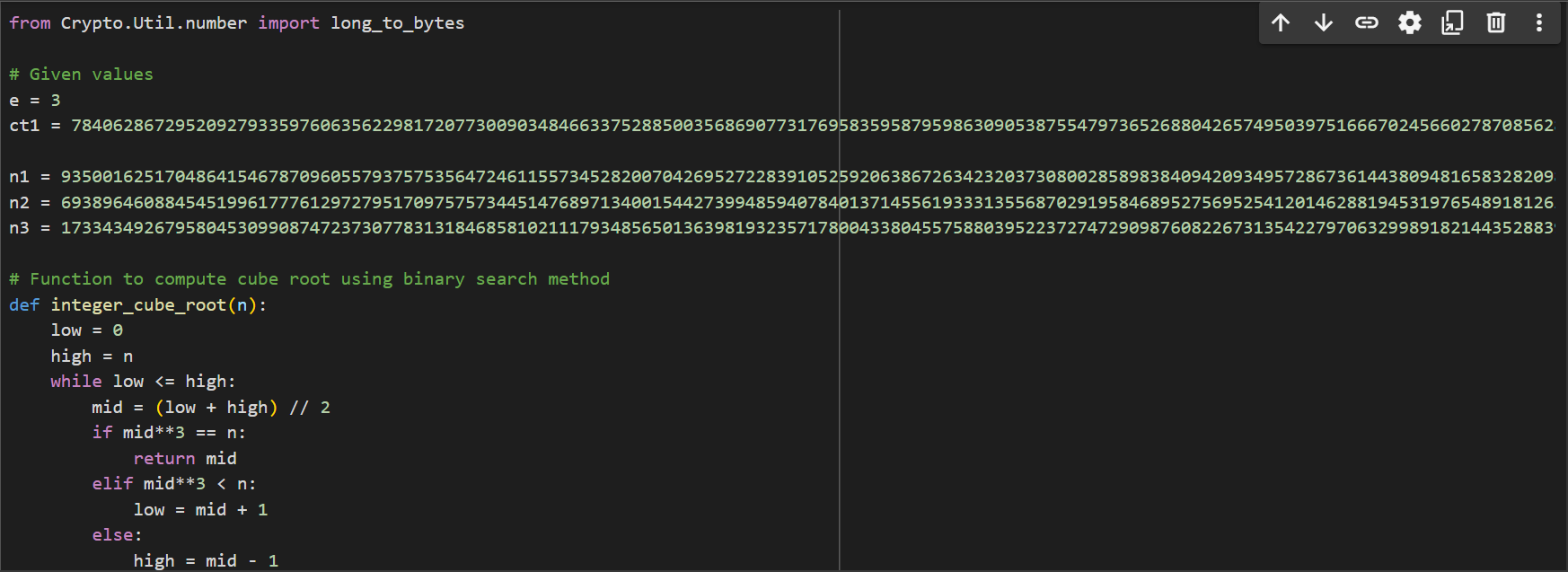
print("Decrypted message from ct3:", plaintext3.decode())

OUTPUT / FLAG

recr{H4s4d\_is\_t0\_g00d}

SCREENSHOTS





A computer screen shot of a program

Description automatically generated