RSA Algo.

TA39 Tanmay Mane.

Code:

```
import math
import random

# Function to compute the modular inverse
def mod_inverse(a, m):
    m0, x0, x1 = m, 0, 1
    while a > 1:
        q = a // m
        m, a = a % m, m
        x0, x1 = x1 - q * x0, x0
    return x1 + m0 if x1 < 0 else x1

# Public key
p = 35
q = 39
n = p * q
print("n =", n)
phi = (p - 1) * (q - 1)

e = random.randrange(1, phi)
print("value of e is", e)
```

```
# Private key
k = 2
d = int(((k * phi) + 1) / e)
print("d =", d)


# Encrypting HI
H = 3
I = 7
m = (H * 100) + I  # Convert HI to a single number
c = pow(m, e, n)
print("c =", c)


# Decrypting
decrypted_m = pow(c, d, n)
decrypted_H = decrypted_m // 100
decrypted_I = decrypted_m % 100
print("Decrypted message: HI =", decrypted_H, decrypted_I)
```

—---

Output:

```
n = 1365
value of e is 496
d = 5
c = 1
```

Decrypted message: HI = 0 1