Diffie Hellman

g=5 p=103

brute-force search: (would not work if p was a large number because it would take too long to brute-force search.

$a \rightarrow ((5^a) \bmod 103) = 10 \rightarrow a = 45$

$b \rightarrow ((5^b) \bmod 103) = 71 \rightarrow b = 67$

$k = B^a \bmod p = 71^{45} \bmod 103 = 31$

$k = A^b \bmod p = 10^{67} \bmod 103 = 31$

- The shared secret is 31.
- my

RSA

Given information: $(e_{Bob}, n_{Bob}) = (17, 266473)$

Find: $M_i = C_i^d \pmod n$

Prime numbers: $439 \times 607 = 266473$

$439 - 1 = 438$ $438 \times 606 = 265428$

$607 - 1 = 606$ $2^2 \times 3^2 \times 73 \times 101 = 265428$

find $\lambda(m)$ is smaller:

$\lambda(265428) = 1800$

$d = 187361$

See code for the rest because it takes too long to do by hand.

```python
e = 17
n = 266473
# Factor
p, q = 439, 607
phi = (p - 1) * (q - 1)
# Compute d
def egcd(a, b):
    if b == 0:
        return (a, 1, 0)
    g, x1, y1 = egcd(b, a % b)
    return (g, y1, x1 - (a // b) * y1)
def modinv(a, m):
    g, x, y = egcd(a, m)
    return x % m
d = modinv(e, phi)
ciphertext = [42750, 225049, 67011, 9062, 263924, 83744, 10951, 156009,
    174373, 125655, 207173, 200947, 227576, 183598, 148747, 211083,
    225049, 218587, 191754, 164498, 225049, 171200, 193625, 99766,
    94020, 223044, 38895, 74666, 48846, 219950, 139957, 77545,
    171672, 165278, 150326, 262673, 164498, 142355, 77545, 171672,
    255299, 5768, 264753, 75667, 261607, 31371, 164498, 140654,
    244325, 140696, 40948, 179472, 168428, 34824, 32543, 30633,
    104926, 190298, 148747, 132510, 42607, 232272, 42721, 188452,
    239228, 50536, 216512, 139240, 78779, 166647, 100152, 261607,
    121165]
# Decrypt each block
def decrypt_block(c):
    m = pow(c, d, n)
    return m
# Convert plaintext integer to two ASCII characters
def decode(m):
    first = m // 256
    second = m % 256
    return chr(first) + chr(second)

message = "".join(decode(decrypt_block(c)) for c in ciphertext)
print(message)
```

Alice encoded her message with two ASCII bytes per block

first = m // 256

second = m % 256

Then combine first and second and continue doing that through each block

The process would fail with larger integers at the factoring of n into p and q. This is because computation of larger numbers is impossible

Alice encoding is insecure because she just follows the basic format of RSA with no randomnessing which makes it easy for people to figure it out.