If you haven't found a helpful website yet, try to use https://en.wikipedia.org/wiki/ElGamal_encryption

A bit of research reveals https://github.com/hellok/CTF/blob/master/crypto/problem/elgamal.py which is the only elgamal implementation, as far as I could find, which uses the variable names the task asks for (except for q)

Using that algorithm, I was able to make a solve script that prints out the values we need:

```python
# part 1
p = 65537
g = 31337
x = 3
h = pow(g, x, p)
m=26729
y=3
c1 = pow(g, y, p)
s = pow(h, y, p)
c2 = s * m % p
print(x,h,m,y,c1,s,c2)
# part 2
q = 47769131098520414182480566228824883 19
g = 313337
h = 22574111718727250262376965985941549 13
x = 42424242424242424242424242424242
c1 = 150535877647806568847436997349134700 0
c2 = 78916253042401334448434134251153776 5
def mod_inverse(a, q):
    t, new_t = 0, 1
    r, new_r = q, a
    while new_r != 0:
        quotient = r // new_r
        t, new_t = new_t, t - quotient * new_t
        r, new_r = new_r, r - quotient * new_r
    if t < 0:
        t = t + q
    return t
s = pow(c1, x, q)
s_inv = mod_inverse(s, q)
print(s, s_inv)
m = (c2 * s_inv) % q
print(f"Recovered message: {m}")
```

The only hard part of this challenge is actually understanding what values does the remote server want.

Made with love by: AndreiCat