1.

$\because x = a, x = b, g^x \equiv h(\bmod p)$

$\therefore g^a \equiv h(\bmod p), g^b \equiv h(\bmod p)$

$\therefore g^a \equiv h(\bmod p), g^b \equiv h(\bmod p), g^{a-b} \equiv 1(\bmod p)$

By Fermat's Little Theorem, the fact that g is a primitive root implies that $p-1$ divides $a-b$, so we have $p \equiv 1(\bmod(a-b)), g^{a-b+1} \equiv g^{1(\bmod p-1)}, a \equiv b(\bmod p-1)$

2.

Python3:

```
import math
def powmod(x, y, p):
    res = 1
    x = x % p
    while (y > 0):
        if (y & 1):
            res = (res * x) % p
        y = y >> 1
        x = (x * x) % p
    return res
def discreteLogarithm(a, b, m):
    n = int(math.sqrt(m) + 1)
    value = [0] * m
    for i in range(n, 0, -1):
        value[powmod(a, i * n, m)] = i
    for j in range(n):
        cur = (powmod(a, j, m) * b) % m
        if (value[cur]):
            ans = value[cur] * n - j
            if (ans < m):
                return ans
    return -1
```

```
18
11
18

Process finished with exit code 0
```

(a)18
(b)11
(c)18

3.

No, S_3 is not commutative group.

4.

$\because b = a^{p-1/q}$

$\therefore b^q = (a^{p-1/q})^q = a^{p-1}, a^{p-1} = 1$

$\therefore b^q = 1$

Q is a prime order b=1 or order b=q, but order b can’t be 1, so order b=q.