Iterative Power (Binary Exponentiation)

$3^{10} = 3^8 * 3^2$ 10: 1010

$3^{19} = 3^{16} * 3^2 * 3^1$ 19: 10011

1. Every number can be written as sum of power of 2, for example 19 = 16 + 2 + 1.
2. We can traverse through all bits of a number (LSB to MSB) in O(log n).

```
res = 1
while n > 0:
        if (n % 2 != 0):
                # Bit is 1
        else:
                # Bit is 0
        n = n/2
```

```
def power(x, n):
        res = 1
        while n > 0:
                if n % 2 != 0:
                        res = res * x
                x = x * x
                n = n // 2
        return res

x = 4
n = 5
print(power(x, n))
```