



Daily Coding Problem #109

Problem

This problem was asked by Cisco.

Given an unsigned 8-bit integer, swap its even and odd bits. The 1st and 2nd bit should be swapped, the 3rd and 4th bit should be swapped, and so on.

For example, 10101010 should be 01010101. 11100010 should be 11010001.

Bonus: Can you do this in one line?

Solution

We can do this by applying a bitmask over all the even bits, and another one over all the odd bits. Then we shift the even bitmask right by one and the odd bitmask left by one.

```
def swap_bits(x):  
    EVEN = 0b10101010  
    ODD = 0b01010101  
    return (x & EVEN) >> 1 | (x & ODD) << 1
```

In one line, that would be:

```
def swap_bits(x):  
    return (x & 0b10101010) >> 1 | (x & 0b01010101) << 1
```

