

BACK

## Swap Adjacent Bits



DESCRIPTION

SOLUTIONS 7450

COMMENTS 34

RE. &gt;

RECOVERY

SCORE: 100/100

You're given an arbitrary 32-bit integer  $n$ . Take its binary representation, split bits into it in pairs (bit number 0 and 1, bit number 2 and 3, etc.) and swap bits in each pair. Then return the result as a decimal number.

## Example

- For  $n = 13$ , the output should be

`swapAdjacentBits(n) = 14`.

$$13_{10} = 1101_2 \rightsquigarrow \{11\}\{01\}_2 \rightsquigarrow 1110_2 = 14_{10}.$$

- For  $n = 74$ , the output should be

`swapAdjacentBits(n) = 133`.

$$74_{10} = 01001010_2 \rightsquigarrow \{01\}\{00\}\{10\}\{10\}_2 \rightsquigarrow 10000101_2 = 133_{10}.$$

Note the preceding zero written in front of the initial number: since both numbers are 32-bit integers, they have 32 bits in their binary representation. The preceding zeros in other cases don't matter, so they are omitted. Here, however, it does make a difference.

## Input/Output

- [execution time limit] 4 seconds (js)
- [input] integer  $n$

*Guaranteed constraints:*

$$0 \leq n < 2^{30}.$$

- [output] integer

## [JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
    console.log("This prints to the console when you Run Tests");
    return "Hello, " + name;
}
```

