

Flipping bits

You will be given a list of 32 bit unsigned integers. Flip all the bits ($1 \rightarrow 0$ and $0 \rightarrow 1$) and return the result as an unsigned integer.

Example

$n = 9_{10}$

$9_{10} = 1001_2$. We're working with 32 bits, so:

$000000000000000000000000000000001001_2 = 9_{10}$

$11111111111111111111111111111110110_2 = 4294967286_{10}$

Return **4294967286**.

Function Description

Complete the *flippingBits* function in the editor below.

flippingBits has the following parameter(s):

- *int n*: an integer

Returns

- *int*: the unsigned decimal integer result

Input Format

The first line of the input contains *q*, the number of queries.
Each of the next *q* lines contain an integer, *n*, to process.

Constraints

$1 \leq q \leq 100$
 $0 \leq n < 2^{32}$

Sample Input

```
3
2147483647
1
0
```

Sample Output

```
2147483648
4294967294
4294967295
```

Explanation

Take *1* for example, as unsigned 32-bits is *00000000000000000000000000000001* and doing the flipping we get *11111111111111111111111111111110* which in turn is *4294967294*.