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 print the result as an unsigned integer.

For example, your decimal input $n = 9_{10} = 1001_2$. We're working with 32 bits, so:

$00000000000000000000000000001001_2 = 9_{10}$
$11111111111111111111111111110110_2 = 4294967286_{10}$

Function Description

Complete the flippingBits function in the editor below. It should return the unsigned decimal integer result.

flippingBits has the following parameter(s):

- n: an integer

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

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

Output Format

Output one line per element from the list with the decimal value of the resulting unsigned integer.

Sample Input 0

```
3
2147483647
1
0
```

Sample Output 0

```
2147483648
4294967294
4294967295
```

Explanation 0

$01111111111111111111111111111111_2 = 2147483647_{10}$
$10000000000000000000000000000000_2 = 2147483648_{10}$
$00000000000000000000000000000001_2 = 1_{10}$
$11111111111111111111111111111110_2 = 4294967294_{10}$
$00000000000000000000000000000000_2 = 0_{10}$
$11111111111111111111111111111111_2 = 4294967295_{10}$

Sample Input 1

```
2
4
123456
```

Sample Output 1

```
4294967291
4294843839
```

Explanation 1

$00000000000000000000000000000100_2 = 4_{10}$
$11111111111111111111111111111011_2 = 4294967291_{10}$
$00000000000000011110001001000000_2 = 123456_{10}$
$11111111111111100001110110111111_2 = 4294843839_{10}$

Sample Input 2

```
3
0
802743475
35601423
```

Sample Output 2

```
4294967295
3492223820
4259365872
```

Author shaka_shadows
Difficulty Easy
Max Score 40
Submitted By 56206

NEED HELP?

View discussions
View editorial
View top submissions

RESOURCES

Binary

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

Download problem statement
Download sample test cases
Suggest Edits

CHOOSE A TRANSLATION

Russian

Explanation 2

$00000000000000000000000000000000_2 = 0_{10}$
$11111111111111111111111111111111_2 = 4294967295_{10}$
$00101111101100011100100101100011_2 = 802743475_{10}$
$11010000001001110001101101001100_2 = 3492223820_{10}$
$00000010000111110011110000001111_2 = 35601423_{10}$
$11111101111000001100001111110000_2 = 4259365872_{10}$

C++

```
#include <bits/stdc++.h>

using namespace std;

// Complete the flippingBits function below.
long flippingBits(long n) {


}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    int q;
    cin >> q;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    for (int q_itr = 0; q_itr < q; q_itr++) {
        long n;
        cin >> n;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        long result = flippingBits(n);

        fout << result << "\n";
    }

    fout.close();

    return 0;
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code