A - Reverse Bits

Given a number, reverse the bits in the binary representation (consider 32-bit unsigned data) of the number, and print the new number formed.

**Input Format**

First line of input contains T - number of test cases. Each of the next T lines contains a number integer N.

**Constraints**

1 <= T <= 100000  
0 <= N <= 109

**Output Format**

For each test case, print the new number formed after reversing the bits, separated by new line.

**Sample Input**

4  
4  
15  
6  
1000000000

**Sample Output**

536870912  
4026531840  
1610612736  
5462492

**Explanation**

**Test Case 1**

Binary Representation of 4: 000...100  
After reversing the bits: 001...000 (536870912)

**Test Case 2**

Binary Representation of 15: 000...1111  
After reversing the bits: 1111...000 (4026531840)

#include <iostream>

using namespace *std*;

unsigned int reverseBits(unsigned int n) {

unsigned int ans = 0;

for (auto i = 0; i < 32; i++)

{

ans = ans << 1;

ans = ans | (n & 1);

n = n >> 1;

}

return ans;

}

int main()

{

*ios\_base*::*sync\_with\_stdio*(false);

*cout*.*tie*(nullptr);

*cin*.*tie*(nullptr);

int t; *cin* >> t;

while (t--)

{

int n; *cin* >> n;

*cout* << reverseBits(n) << "\n";

}

return 0;

}