Print Big Binary Numbers

Some answers for some problems could be huge binary numbers. In order to check the computation, one could ask you the sum of its digits. With a little base, the answer is a small number too, but not with a bigger base.

XerK would like to avoid precomputed results and wish check you've computed his huge numbers. Here's a problem that check computation of a big number N. A tutorial edition exists without language restrictions.

Let define the function CHK(N, B):
Input : N a big number in binary representation, B a power of two. Consider N as a base B number.
Output : the sum of its digits in this base.

Example :with B=2^8, 12345678 = 78 + 97*B + 188*B*B, so CHK(12345678, B) = 78 + 97 + 188

This should be easily computed with few bitwise-AND, bitshifts and additions.

Input

The input begins with the number T of test cases in a single line.
In each of the next T lines there are four integers A, B, C, D, given in base 10.

Output

For each test case :
* compute N = (A^B) XOR (C^D).
* print CHK(N, 2^16384) as a base 10 number.
(^ denote the power, and XOR the bitwise operator)

Example

Input:
2
7 3 5 4
1234 5678 9012 4444

Output:
806
11942041587942321477 99<...snip...>9938532444216215551948305

Explanations

For test case 1:
7^3 = 343, 5^4 = 625, 343 XOR 625 = 806, CHK(806, 2^16384) = 806.

For test case 2:
You have to output all 4933 digits of the result.

Constraints

1 < T <= 321
1 < A, B, C, D <= 10^4

Edit 2017-02-11, after compiler update ; new TL.