Project Euler #97: Large non-Mersenne prime

This problem is a programming version of Problem 97 from projecteuler.net

The first known prime found to exceed one million digits was discovered in 1999, and is a Mersenne prime of the form $2^{6972593} - 1$; it contains exactly 2,098,960 digits. Subsequently other Mersenne primes, of the form $2^p - 1$, have been found which contain more digits.

However, in 2004 there was found a massive non-Mersenne prime which contains 2,357,207 digits: $28433 \times 2^{7830457} + 1$.

Now we want to learn how to calculate some last digits of such big numbers. Let's assume we have a lot of numbers $A \times B^C + D$ and we want to know last 12 digits of these numbers.

Input Format

First line contains one integer T - the number of tests.
T lines follow containing 4 integers (A, B, C and D) each.

Constraints
$1 \leq T \leq 500000$
$1 \leq A, B, C, D \leq 10^9$

Output Format

Output exactly one line containing exactly 12 digits - the last 12 digits of the sum of all results. If the sum is less than 10^{12} print corresponding number of leading zeroes then.

Sample Input

```
1
2 3 4 5
```

Sample Output

```
000000000167
```

Explanation

$2 \times 3^4 + 5 = 2 \times 81 + 5 = 162 + 5 = 167$