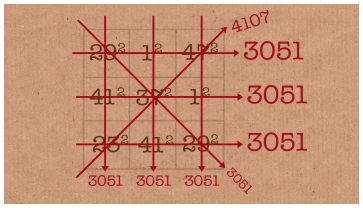
Y It's a Kind of Magic

Time limit: 4s

Everyone knows that a 3×3 magic square must meet two criteria:

1. All nine numbers must be positive and distinct.
2. The sums along all rows, columns, and diagonals are equal.



The Parker Square. © Brady Haran, used with permission

Everyone, except maybe Matt Parker[1]. He wants to create a magic square of squares, that is, a magic square that also meets a third criterion:

3. Each number is a square of a positive integer.

His "result" can be seen in the picture in the corner. As you may notice, his square is not that magic... Not only do most of the values appear twice, it also has a diagonal with the wrong sum. To be honest, apart from containing non-square values, there is not much that could make this square worse. Well, at least he tried!

But that is all in the past. After finding the *Parker Square*, he decided to completely ignore property 3 from now on and to instead give property 2 a new twist. He now considers multiplicative magic squares, which are exactly like normal magic squares except that the *products* along all rows, columns, and diagonals have to be equal, instead of the sums. Who knows, Matt might even manage to find a proper multiplicative magic square in the future!

With this definition at hand, Matt wrote some terrible Python code – his words, not ours – to count the number of multiplicative magic 3×3 squares with product at most n. As you may have guessed by now, his code is way too slow. Therefore, we task you to do the same, just more efficiently. Given an integer n, count the number of multiplicative magic 3×3 squares with product at most n.

Input

The input consists of:

- One line with an integer t ($1 \leq t \leq 10^5$), the number of test cases.
- t lines, each with an integer n ($1 \leq n \leq 10^{18}$), the maximum product.

Output

For each test case, output the number of multiplicative magic squares with product at most n.

[1] Recreational mathematician, author, comedian, YouTube personality, and science communicator.

Sample Input 1	**Sample Output 1**
3 500 1000 3000	8 16 56