

RReeppeettiittiioonn

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 512 megabytes

As a girl born on November 11th (or 11/11), SS loves repetition. For example, her favorite English word is *bookkeeper*, her favorite letter is *W* (double-u, double vé, or whatever you call it), her favorite animals are the cat (*kitten!*) and the rabbit (*bunny!*), her favorite programming language is *C++*, her favorite parser is *LL(1)*, her favorite optimization algorithm is branch and bound (*BB*), her favorite unit is millimeter (*mm*), her favorite game is *LLLL*, and her favorite winning hand in Mahjong is Seven Pairs. She plans to release her new songs “Silent-night Satellite” and “Siesta Stella” soon, so she wants to select a “pretty” number to decorate the cover of her new songs.

SS observes this world from a higher perspective, which means she considers an integer concurrently in all standard positional numeral systems, starting from the binary numeral system. SS then calls a positive integer n (b, p) -tidy (with $b, p \geq 2$) if and only if:

- In base- b representation, n is represented as $\overline{d_{k \cdot p - 1} d_{k \cdot p - 2} \dots d_1 d_0}$ for some positive integer k and integers $0 \leq d_i < b$, such that for each $0 \leq i < k$, $d_{i \cdot p} = d_{i \cdot p + 1} = \dots = d_{(i+1) \cdot p - 1}$.

For example, 2233 and 3355 are $(10, 2)$ -tidy, 1111 is both $(10, 2)$ -tidy and $(10, 4)$ -tidy, and $6\,737\,151 = (66\text{CCFF})_{16}$ is $(16, 2)$ -tidy. Furthermore, SS defines the *tidiness* of a positive integer n as the number of ordered pairs (b, p) such that n is (b, p) -tidy.

SS has given you an integer n . Your task is to determine the *tidiness* of n .

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 1000$). The description of the test cases follows.

The only line of each test case contains a single integer n ($1 \leq n \leq 10^{12}$) — the integer SS has given you. It is guaranteed that the sum of n over all test cases does not exceed 10^{12} .

Output

For each test case, output a single integer — the *tidiness* of n .

Example

standard input	standard output
10	0
1	0
2	2
115	4
1111	5
2233	5
3355	24
191970	9
6737151	17
102934760424	144
618111100000	

Note

In the third test case, $115 = (55)_{22} = (11)_{114}$. Therefore, the tidiness of 115 is 2.

In the fourth test case, 1111 is $(10, 2)$ -*tidy*, $(10, 4)$ -*tidy*, $(100, 2)$ -*tidy*, and $(1110, 2)$ -*tidy*.