

1. In the theory of numbers, square free numbers have a special place. A square free number is one that is not divisible by a perfect square (other than 1).

Problem Description

In the theory of numbers, square free numbers have a special place. A square free number is one that is not divisible by a perfect square (other than 1). Thus 72 is divisible by 36 (a perfect square), and is not a square free number, but 70 has factors 1, 2, 5, 7, 10, 14, 35 and 70. As none of these are perfect squares (other than 1), 70 is a square free number.

For some algorithms, it is important to find out the square free numbers that divide a number. Note that 1 is not considered a square free number.

In this problem, you are asked to write a program to find the number of square free numbers that divide a given number.

Input

The only line of the input is a single integer N which is divisible by no prime number larger than 19

Output

One line containing an integer that gives the number of square free numbers (not including 1)

Constraints

$N < 10^9$

Time Limit (secs)

Examples

Example 1

Input

20

Output

3

Explanation

N=20

If we list the numbers that divide 20, they are

1, 2, 4, 5, 10, 20

1 is not a square free number, 4 is a perfect square, and 20 is divisible by 4, a perfect square. 2 and 5, being prime, are square free, and 10 is divisible by 1,2,5 and 10, none of which are perfect squares. Hence the square free numbers that divide 20 are 2, 5, 10. Hence the result is 3.

Example 2

Input

72

Output

3

Explanation

N=72. The numbers that divide 72 are

1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72

1 is not considered square free. 4, 9 and 36 are perfect squares, and 8,12,18,24 and 72 are divisible by one of the. Hence only 2, 3 and 6 are square free. (It is easily seen that none of them are divisible by a perfect square). The result is 3

2. Given N number of x's, perform logic equivalent of the above Java code and print the output

Problem Description

```
Scanner sc = new Scanner(System.in);
```

```
long sum = 0;
```

```
int N = sc.nextInt();
```

```
for (int i = 0; i < N; i++) {
```

```
    final long x = sc.nextLong(); // read input
```

```
    String str = Long.toString((long) Math.pow(1 << 1, x));
```

```
    str = str.length() > 2 ? str.substring(str.length() - 2) : str;
```

```
    sum += Integer.parseInt(str);
```

```
}
```

```
logger.debug(sum%100);
```

Given N number of x's, perform logic equivalent of the above Java code and print the output

Input

First line contains an integer N

Second line will contain N numbers delimited by space

Output

Number that is the output of the given code by taking inputs as specified above

Constraints

$$1 \leq N \leq 10^7$$

$$0 \leq x \leq 10^{18}$$

Time Limit (secs)

1

Examples

Example 1

Input

4

8 6 7 4

Output

64

Example 2

Input

3

1 2 3

Output

14