

Problem 5 – Search in Bits

You are given a list of **N** numbers and a number **S**.

Count the occurrences of the most right **4 bits** of **S** in the most right **30 bits** in **every given number**.

For example there are 3 occurrences of the 4 most right bits of 9 in the number 1177.

The 4 most right bits of 9 are 1001 and the 30 most right bits in the number 1177 are 0000000000000000000010010011001.

The occurrences are:

- 00000000000000000000100110011001
- 0000000000000000000010010011001
- 0000000000000000000010010011001

It is allowed for one occurrence to be part of another one (like the first and the second one in the above example).

Input

The input data should be read from the console.

On the first line there will be the number S .

On the second line there will be the number **N**.

On each of the next **N** lines there will be a number from the list.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data should be printed on the console.

On the only output line print the number of occurrences found.

Constraints

- **N** will be an integer between 1 and 100, inclusive.
- **S** will be an integer between 0 and 15, inclusive.
- All numbers will be integers between 0 and 1073741823, inclusive.
- Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

Examples

Example input	Example output	Explanation
9 1 1177	3	The 4 bits of S are 1001 There are 3 occurrences in the number: 00000000000000000000 <u>1001</u> 0011001 00000000000000000000100 <u>1001</u> 1001 000000000000000000001001001 <u>1001</u>
4 3 4 536873260 536870912	3	The bits of S are 0100 There is 1 occurrence in the first number: 00000000000000000000000000 <u>0100</u> There are 2 occurrences in the second number:

		100000000000000000 <u>0100</u> 100101100 100000000000000000010 <u>0100</u> 101100 There are no occurrences in the third number. 1000000000000000000000000000000000
--	--	---