Problem A: Goodbye

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

You are to write a very basic program. It should just output **"Goodbye, real life!"** on a single line, no matter what the input.

Sample input and output

Input	**Output**
12345	Goodbye, real life!

Problem B: Binary

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Your private cookie machine just quit service. You suspect its software (which you wrote yourself) to be at fault. To find the bug, you first want to know what the machine thinks how many cookies were in production when it broke down. To obtain that number, you need to analyze the *production mask*, an array of bits, each indicating the status of one of the production units (1 = producing, 0 = waiting), which you can only access as an integer number.

Input

The input consists of a single integer p ($0 \leq p \leq 2^{64} - 1$), the integer representation of the production mask.

Output

Print a single line containing the number of active production units according to the production mask you were given.

Sample input and output

Input	**Output**
255	8

Problem C: Filter

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Given a list of integers, you are to find those that are either of the form $a = 2^r s$ with $r \geq 1$, s odd or of the form $a = 3^t u$ with $t \geq 1$, u not divisible by 3, but not of both forms simultaneously.

Input

The input consists of

- one line containing N ($1 \leq N \leq 10^5$) – the length of the list
- one line giving the integers $a_1, ..., a_N$ ($|a_i| \leq 10^9$) the list consists of.

Output

Filter out all numbers not fulfilling the condition described above and print the result (i.e. the list of all given numbers that satisfy the condition, in the order you received them).

Sample input and output

Input	**Output**
4 28 17 54 -81	28 -81

Problem D: Searching

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

First you are given a list of integers, then you have to respond to some queries, each of them asking for all occurrences of a certain number in the list.

Input

The input consists of

- one line containing N ($1 \leq N \leq 10^4$) – the length of the list – and M ($1 \leq M \leq N$) – the number of queries
- one line giving the integers $a_1, ..., a_N$ ($|a_i| \leq 10^9$) the list consists of
- M lines each containing a query consisting of a single integer – the number whose occurrences you should locate.

Output

For each of the queries, output the (1-based) indices of all occurrences of the requested number in the list you were given, separated by spaces. If the requested number doesn't occur at all, output "None!" to answer the query.

Sample input and output

Input	**Output**
5 2 32 17 -5 32 12 115 32	None! 1 4

Problem E: Powers

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

You are to write a program that can raise any integer a to any positive power n, i.e. calculate a^n.

Input

The input consists of

- one line containing T ($1 \leq T \leq 1000$) – the number of testcases
- T lines containing the testcases, i.e. numbers a and n ($1 \leq |a|, n \leq 10^9$).

It is guaranteed that $|a^n| \leq 2^{63} - 1$ (i.e. that the result fits in **long long**) for all testcases.

Output

For each of the testcases, output the result a^n in a separate line.

Sample input and output

Input	**Output**
2 -2 3 3 5	-8 243