Problem A. A harder xor problem

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

You are given an integer N. Your task is to compute the cumulative bitwise XOR of all integers from 1 to N. In other words, calculate:

$$1 \oplus 2 \oplus \cdots \oplus N$$

where $\oplus$ denotes the bitwise XOR operation.

Input

The input consists of multiple test cases:

The first line contains a single integer T, the number of test cases.

Each of the next T lines contains a single integer N.

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^{18}$

Output

For each test case, output a single line containing the result of the cumulative XOR from 1 to N.

Examples

Standard Input	Standard Output
4 1 2 3 4	1 3 0 4

Note

For $N = 1$, $1 = 1$

For $N = 2$, $1 \oplus 2 = 3$

For $N = 3$, $1 \oplus 2 \oplus 3 = 0$

For $N = 4$, $1 \oplus 2 \oplus 3 \oplus 4 = 4$

This page is intentionally left blank

Problem B. K sum

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

You are given an array $a_1, a_2, \ldots, a_n$ of length n and an integer k. We define the score of a subarray as the square of its sum. For example, if the subarray is $[1, 2, 3, 4]$, then its score is $(1+2+3+4)^2 = 100$.

For each $i = 1, 2, \ldots, n$, you are asked to find s_i, which is the maximum score you can obtain from a subarray that ends at i and has a length of no more than k.

Since the array size is large, you will generate the array as following:

- $a_1 = y$
- $a_i = (p \cdot a_{i-1} + q) \bmod m$, with $2 \leq i \leq n$

Input

The first input line has two integers n, k: the length of the sequence and.

The second input line has four integers y, p, q, m.

- $1 \leq n, k \leq 2 \cdot 10^7$
- $-100 \leq y, p, q \leq 100$
- $1 \leq m \leq 100$

Output

Print one integer, $s_1 \oplus s_2 \oplus \cdots \oplus s_n$, where $\oplus$ denotes the bitwise XOR operation.

Examples

Standard Input	Standard Output
5 3 -5 -3 5 9	96
89733 37928 -82 -17 18 98	1215940704
19387465 8038834 21 3 7 99	25117131567660229

Note

In sample input 1, array $a = [-5, 2, -1, 8, -1]$.

For $i = 1$, the maximum score is $(-5)^2 = 25$.

For $i = 2$, the maximum score is $[(-5) + 2]^2 = (-3)^2 = 9$

For $i = 3$, the maximum score is $[(-5) + 2 + (-1)]^2 = (-4)^2 = 16$

For $i = 4$, the maximum score is $[2 + (-1) + 8]^2 = 9^2 = 81$

For $i = 5$, the maximum score is $[8 + (-1)]^2 = 7^2 = 49$

Since $25 \oplus 9 \oplus 16 \oplus 81 \oplus 49 = 96$, you should print 96 as your output.

This page is intentionally left blank

Problem C. Line Permutation

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

There are n functions, each of the form $f(x) = ax + b$. Your task is to determine a permutation $P = (f_1(x), f_2(x), \ldots, f_n(x))$ of the functions such that $\max(f_1(1), f_2(2), \ldots, f_n(n))$ is minimized.

Input

The first input line contains an integer n: the number of functions.

Each of the next n lines contains two integers, a and b, representing the coefficients of a function.

- $1 \leq n \leq 10^5$
- $1 \leq a, b \leq 10^9$

Output

Print a single number: $\max(f_1(1), f_2(2), \ldots, f_n(n))$, where $(f_1(x), f_2(x), \ldots, f_n(x))$ is the permutation described above.

Examples

Standard Input	Standard Output
5 4 6 3 9 5 3 3 6 9 1	21

Note

We can reorder the function to

$f_1(x) = 9x + 1$

$f_2(x) = 5x + 3$

$f_3(x) = 4x + 6$

$f_4(x) = 3x + 9$

$f_5(x) = 3x + 6$

Then, $\max(f_1(1), f_2(2), f_3(3), f_4(4), f_5(5)) = \max(10, 13, 18, 21, 21) = 21$. It can be proven that 21 is the minimum you could obtain in this test case.

This page is intentionally left blank

Problem D. K common divisor

Time limit:	please refer to DOM Judge
Memory limit:	please refer to DOM Judge

You are given an integer sequence a of length n and a positive integer k. Define the set S as:

$$S = \{x : \text{there are at least } k \text{ elements in } a \text{ that are divisible by } x\}$$

Find $\sum_{x \in S} x$.

Input

The first input line contains two integers n and k. The second line contains n integers $a_1, a_2, \ldots, a_n$.

- $1 \le k \le n \le 2 \cdot 10^6$
- $1 \le a_i \le 2 \cdot 10^6,\ i = 1, 2, \ldots, n$

Output

Print a single integer: $\sum_{x \in S} x$.

Examples

Standard Input	Standard Output
10 4 7 9 2 3 27 10 30 42 6 18	12

Note

In the sample test case, there are four numbers that has at least $k = 4$ elements in a that is divisible by x:

- $1 : 7, 9, 2, 3, 27, 10, 30, 42, 6, 18$
- $2 : 2, 10, 30, 42, 6, 18$
- $3 : 9, 3, 27, 30, 42, 6, 18$
- $6 : 30, 42, 6, 18$

The sum is $1 + 2 + 3 + 6 = 12$.

This page is intentionally left blank

Problem E. Quartile

Time limit:	please refer to DOM Judge
Memory limit:	please refer to DOM Judge

You are given an initially empty sequence S.

Total of n numbers will be inserted into S one by one. After each insertion, you need to determine the value at the lower quartile position in the sorted sequence of S.

Input

The first line contains an integer n, the number of integers to be inserted into S.

Each of the next n lines contains a single integer representing a number to be inserted into S.

- $1 \leq n \leq 10^5$ —the number of integers to be inserted.
- $1 \leq$ each inserted number $\leq 10^9$.

Output

For each insertion, immediately output the quartile of the numbers currently in S.

Examples

Standard Input	Standard Output
5 1 2 3 4 5	1 1 1 2 2
5 5 4 3 2 1	5 4 3 3 2

Note

Suppose there are m numbers $a_0, a_1, \ldots, a_{m-1}$ sorted in non-decreasing order. The quartile of these numbers is defined as the element at the position $\lfloor \frac{m}{4} \rfloor$ (0-indexed).

This page is intentionally left blank

Problem F. Sticks

Time limit:	please refer to DOM Judge
Memory limit:	please refer to DOM Judge

George took sticks of the same length and cut them randomly into small parts. Now he wants to return sticks to the original state, but he forgot how many sticks he had originally and how long they were originally. Please help him and design a program which computes the smallest possible original length of those sticks.

Input

The input file contains several blocks of 2 lines, each block represents a test case. The first line contains an integer n, which is the number of sticks parts after cutting. The second line contains n integers a_i, which are the lengths of those parts.

The last line of the file contains a single 0.

- $1 \leq n \leq 64$
- $1 \leq a_i \leq 50$
- $1 \leq$ number of test cases ≤ 150

Output

The output contains the smallest possible length of original sticks, one per line

Examples

Standard Input	Standard Output
9 5 2 1 5 2 1 5 2 1 4 1 2 3 4 0	6 5

Note

In the first test case, we can form sticks in length of 6 by $5 + 1$, $5 + 1$, $5 + 1$, and $2 + 2 + 2$.

In the second test case, we can form sticks in length of 5 by $1 + 4$ and $2 + 3$.

Note that you must use all stick parts.

This page is intentionally left blank