Problem A. Substrings and Subsequences

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Vasya got tired of confusion between substrings and subsequences. So he wants to know what is the probability for a randomly generated string s that the set of all its different subsequences equals to the set of all its different substrings.

Vasya generates strings in the following way: each of n characters is generated independently from the same distribution. Probability that i-th character of the string equals to j-th letter of English alphabet is equal to

$$\frac{w_j}{\sum_{k=1}^{m} w_k}.$$

Input

The first line of the input contains integer T ($1 \le T \le 1000$) — the number of testcases.

Description of each test case consists of two lines. The first line contains two integers n and m ($1 \le n \le 10^9, 1 \le m \le 26$) — the number of characters in the string and the number of letters used to generate strings.

The second line of each test case contains m integers w_i ($1 \le w_i, \sum_{i=1}^{m} w_i \le 10^9$).

Output

If the answer for i-th testcase equals irreducible fraction $\frac{P}{Q}$, then print $P \cdot Q^{-1} \bmod (10^9+7)$. Print answers for different testcases in separate lines.

Examples

standard input	standard output
3 3 1 5 2 5 1 2 3 4 5 5 5 1 2 3 4 5	1 1 819476549

Problem B. Bus stop

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

Your bus stop belongs to n different bus routes. Every route has interval t_i — time in minutes between previous bus departure and next bus arrival. The time bus spends staying at the stop is negligible. Schedules of different routes are independent and aren't known to you. All routes suit you. How long in average will you wait for a bus if you come to the stop in a random moment of time?

In other words, you need to calculate the expected value of minimum of n independent uniformly distributed on $[0; t_i]$ variables.

Input

The first line contains one integer n ($1 \le n \le 10^5$) — the number of bus routes.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le 10^9$) — intervals in minutes for each bus route.

Output

Print one real number — the expectation of time, in minutes, after which comes the first bus.

Your answer will be considered correct if its absolute or relative error will not exceed 10^{-8}.

Examples

standard input	standard output
2 10 20	4.166666666667
5 10 10 20 20 30	2.402777777778

Problem C. Collection of sets

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given set $A = \{1, 2, \ldots, n\}$ and collection of its subsets L each having the same size k, consider the following process:

1. Assign weight 1 or 2 equiprobably independently to each element of A.
2. Calculate the weight of every subset in L as a sum of weights of its elements.
3. Assignment of weights is considered *good* if only one subset in L has the minimal weight.

You are given set $A = \{1, 2, \ldots, n\}$ and integer k, denoting the size of every subset in L. Your task is to build such collection L, that the probability of the good assignment is less than $\frac{1}{100}$.

For example, $A = \{1, 2\}$, $L = \{\{1\}, \{2\}\}$. Assignments $\{1, 2\}$ and $\{2, 1\}$ are good, while $\{1, 1\}, \{2, 2\}$ aren't good, because weights of subsets are equal and both of them are minimal. So probability of good assigment equals $\frac{1}{2}$.

Input

The only line of the input file contains two integers n, k.

In sample $n = 2, k = 1$, and any correctly formatted output will be judged as OK. Note that there is no collection satisfying the requirements from the statement for that case.

In tests $n = 14$, $2 \le k \le 12$ and it is guaranteed that solution exists.

Output

On the first line print integer m — the number of subsets in L.

Next m lines contain descriptions of subsets. i-th of them contains k distinct integers between 1 and n separated by spaces, denoting its elements.

All subsets in the output must be distinct.

Examples

standard input	standard output
2 1	2 1 2

Problem D. Decomposition

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given a graph P with n vertices and m edges. You should decompose it to cartesian product of some graph Q and boolean cube graph with **maximal power** k.

Graph of boolean cube of power k is a graph with 2^k vertices numerated from 0 to $2^k - 1$. Two vertices are adjacent if and only if their binary representations (with leading zeros) differ in exactly one position.

Cartesian product $G \square H$ of graphs G and H is a graph such that

1. the vertex set of $G \square H$ is the Cartesian product of sets $V(G) \times V(H)$
2. any two vertices (u, u') and (v, v') are adjacent in $G \square H$ if and only if either
   - $u = v$ and u' is adjacent with v' in H, or
   - $u' = v'$ and u is adjacent with v in G.

Input

First line contains 2 integers n, m ($1 \le n \le 2 \cdot 10^5, 0 \le m \le 2 \cdot 10^5$) — the number of vertices and the number of edges in the graph P.

Each of next m lines contains 2 integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$), describing i-th edge.

It's guaranteed that there is at most one edge between any pair of vertices.

Output

First line should contains one integer k.

If $k > 0$ then output another n lines. i-th of them should contain one integer $1 \le c_i \le \frac{n}{2^k}$ and a k-bit string, denoting the vertex number in graph Q and description of boolean cube vertex which was used to product i-th vertex of graph P.

In other words, you should describe the bijection between the vertices of the input graph and pairs (u, v) where u is the vertex of the graph Q and v is the vertex of the boolean cube graph.

Examples

standard input	standard output
4 4 1 3 3 2 2 4 4 1	2 1 00 1 11 1 01 1 10
6 9 1 4 2 5 3 6 1 2 1 3 2 3 4 6 5 6 4 5	1 1 0 3 0 2 0 1 1 3 1 2 1
3 3 1 2 2 3 1 3	0

Note

In mathematics, a Cartesian product is a mathematical operation which returns a set (or product set or simply product) from multiple sets. That is, for sets A and B, the Cartesian product $A \times B$ is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$. (Wikipedia)

Problem E. Easy Everest

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

There are n climbers on the south slope of Everest, staying in points with heights $h_1, h_2, \ldots, h_n$, and also n climbers on the north slope, staying in points with heights $g_1, g_2, \ldots, g_n$.

Let's call an arrangement of the climbers *safe* if for each possible height the number of climbers situated in points with such height on both slopes are equal. The climbers want to take a safe arrangement to ride out the storm. In order to do this, every climber can change his position on his slope (he could not go to other slope). To change the height of his position from a to b, the climber has to spend $|b - a|$ units of energy. Now, since the hike is definitely going to be difficult, the climbers want to spend the minimum total amount of energy to take a safe arrangement. Your task is to calculate that amount.

Input

The first line of the input contains integer n ($1 \le n \le 3 \cdot 10^5$) — the number of climbers.

The second line contains n integers h_i ($1 \le h_i \le 10^9$) — the heights of positions of the climbers on the south slope.

The third line contains n integers g_i ($1 \le g_i \le 10^9$) — the heights of positions of the climbers on the north slope.

Output

Output one line with the minimum total amount of energy the climbers need to spend to take a safe arrangement.

Examples

standard input	standard output
2 1 7 4 3	5
3 1 2 2 1 2 1	1

Problem F. Sum of divisors

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

We call the integer N K-*expressible* if it can be expressed as a sum of K divisors of N (not necessarily different). Find the number of K-expressible integers between A and B inclusive.

Input

The first line contains integer T ($1 \le T \le 5 \cdot 10^4$) — the number of testcases.

Next T lines contain the description of testcases. Each of them contains three integers A, B, K ($1 \le A \le B \le 10^{18}$, $2 \le K \le 7$), separated by spaces.

Output

Print T lines, containing the answers for the corresponding testcases.

Examples

standard input	standard output
3 1 5 3 5 10 2 4 6 4	2 3 2

Note

In the first testcase, the numbers 3 and 4 are 3-expressible: $3 = 1 + 1 + 1$, $4 = 1 + 1 + 2$.

In the third testcase, the numbers 4 and 6 are 4-expressible: $4 = 1 + 1 + 1 + 1$, $6 = 1 + 1 + 1 + 3$.

Problem G. Guess sinus

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

This is an interactive problem. Jury chose a real number a ($|a| \leq 10^9$) with no more than 5 digits after decimal point and you need to guess it. You could send jury queries of two types:

1. Send real number x, jury will send you $sign(sin(a \cdot x))$.
2. Send real number y trying to guess number a.

You could send not more than 200 queries of the first type and only one query of the second type. Your answer will be considered correct if its absolute or relative error will not exceed 10^{-6}.

Input

For each your query of first type a single line will be printed, containing one number: 0, 1 or -1.

There will be no response for the second type query.

Output

In each line print one your query.

Print `"? x"` to get $sign(sin(a \cdot x))$ or print `"! y"` if you think that answer is y.

You may use no more than 200 queries of the first type.

Your program should terminate immediately after it sends a query of the second type.

Examples

standard input	standard output
1 -1	? 1 ? -1 ! 1

Note

$sin(a \cdot x)$ will be calculated using standard C++ functions from math.h and `double` datatype.

```
sign(double x) = x > 0 ? 1 : (x < 0 ? -1 : 0)
```

Problem H. Matrices dot product

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

You are given $n \times n$ matrix S and integer k, $2^k \le n$. Consider the following matrix recurrent:

$$T_0 = \begin{pmatrix} 1 \end{pmatrix}$$

$$T_{i+1} = \begin{pmatrix} T_i \cdot a_{00} & T_i \cdot a_{01} \\ T_i \cdot a_{10} & T_i \cdot a_{11} \end{pmatrix}$$

where

$$A = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$$

is a constant 2×2 matrix.

Find the minimum and maximum values of dot products of matrix T_i and every continuous $2^i \times 2^i$ submatrix of S for each i such as $0 \le i \le k$.

Dot product of matrices $\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$ and $\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix}$ is $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot b_{ij}$.

Input

First line contains 2 integers n, k ($1 \le n \le 1000, 1 \le 2^k \le n$).

Each of the next n lines contains n integers, j-th integer of i-th line is S_{ij} ($-10^9 \le S_{ij} \le 10^9$).

Each of the next 2 lines contains 2 integers in the same format, denoting matrix A. Elements of A do not exceed 2 by absolute value.

Output

Print $k+1$ lines. $(i+1)$-th line should contain 2 integers — the minimum and the maximum dot product of continuous submatrices of S and T_i.

Examples

standard input	standard output
3 1 9 -3 13 0 10 3 10 -10 0 1 -1 -1 1	-10 13 -30 22

Problem I. I18n

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

$i18n$ compression for a string $s_1 \dots s_n$ ($n \geq 4$) is a string $s_1 + to\_string(n-2) + s_n$, where $to\_string(k)$ is a string containing decimal represenation of number k. For example, $i18n(abxcd) = a3d$.

Your task is to minimize the length of given text using $i18n$ compression for some words (the remaining words will not change). You should compress words in such way that for every compressed word x there is exactly one uncompressed word y before x such as $i18n(y) = i18n(x)$ and y must be equal to x.

Input

The only line of input contains space separated words consisting of lowercase Latin letters. The length of input file doesn't exceed 10^5 characters.

Output

Print one line with space separated compressed text from input.

Examples

standard input	standard output
internationalization bla bla bla abcd acbd bcda abcd internationalization internationalization	internationalization bla bla bla abcd acbd bcda abcd i18n i18n
a bb ccc dddd ccc bb a	a bb ccc dddd ccc bb a

Note

In the first test:

- *bla* is not compressed because it's length is less then 4,
- *acbd* is not compressed because there is *abcd* before it, $i18n(abcd) = i18n(acdb)$, but $abcd \neq acbd$,
- the second one *abcd* is not compressed because there are two uncompressed words with the same $i18n$ before it: *abcd* and *acbd*.

Problem J. Segment Sort

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are a scientist from famous research centre Oskolkovo. You invented a quantum algorithm which could sort an array of length n using $\lceil\sqrt{n}\rceil$ quantum operations (don't ask us how it is possible). You wanted to prove the power of your invention by sorting an array of n integers, but suddenly you found out that your competitors from MIT had already published an article about a similar algorithm with the same complexity. So you decided to cheat a little bit and allowed your algorithm not only to sort the whole array, but to sort some subarrays (if it helps to reduce the total number of quantum operations). These operations are applied consequently, and the subarrays sorted during them may intersect.

Input

The first line contains one integer n ($1 \le n \le 10^5$) — the number of elements in the array.

Next line contains n integers a_i ($0 \le a_i \le 10^9$) — the elements of the array.

Output

Output one line with minimal number of quantum operations needed to sort given array by sorting its subarrays, given that cost of sorting a subarray of length l is $\lceil\sqrt{l}\rceil$.

Examples

standard input	standard output
5 5 1 2 3 1	3
5 1 2 1 3 5	2

Problem K. Tree generation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

After the last Codeforces round Vasya has become obsessed with creation of his own problems. Vasya has a poor imagination, but he hates cactuses, so he decided to create a problem about tree. He easily came up with just another task where you need to run two dfs on the tree, but the next problem he encountered is to generate tests for it.

Vasya's most favorite function from testlib.h is wnext. For our purposes, in this problem we simplify its definition, and it works in the following way:

```
int wnext(int a, int b, int type) {
   int result = next(a, b);
   for (int i = 0; i < type; i++)
        result = max(result, next(a, b));
   for (int i = 0; i < -type; i++)
        result = min(result, next(a, b));
   return result;
}
```

Function $next(a, b)$ returns the uniformly distributed random integer in range $[a; b]$.

Vasya came up with the following way to generate tests: he generates tree with $n = 10000$ vertices. Vertices of the tree are numbered from 0 to $n - 1$. He chooses random integer *type* in range $[-4; 4]$, iterates over vertices of the tree from vertex 1 to vertex $n-1$, adding an edge between vertex i and vertex $wnext(0, i - 1, type)$. After that, Vasya randomly shuffles the vertices of the tree. This procedure can be described by the following code:

```
int n = 10000;
int p[n];
for (int i = 0; i < n; ++i) {
    p[i] = i;
}
shuffle(p, p + n);
int type = next(-max_param, max_param);
for (int i = 1; i < n; ++i) {
    add_edge(p[i], p[wnext(0, i - 1, type)]);
}
```

Here p is a random permutation of integers from 0 to $n - 1$.

Vasya generated 100 testcases for his problem in such way. Accidentally, he forgot to mark each testcase by the value of the parameter *type* used in it. Given the testcases generated by Vasya, help him to recover the parameter values for at least 90% of testcases. Parameters for all testcases are choosen independently.

Input

The first line contains one integer T — the number of testcases, which is always equal 100.

Next T lines contain the decsription of testcases. Each line starts with integer n — the number of vertices in the tree, which is always equal 10000. The next $n - 1$ intgers $p_1, p_2, \ldots, p_{n-1}$ $(0 \le p_i \le n - 1, p_i \ne i)$ denote the edges of the tree (i, p_i). Note that the vertices were shuffled by Vasya, so it's not guaranteed that $p_i < i$. It's guaranteed that edges (i, p_i) form a tree on n vertices.

In the sample, just to show the format, $T = 2$, $n = 3$. The sample was not generated as shown above, and your solution will not be tested on the sample. The first test is different from the sample.

Output

Output T integers, one per line, — the values of the parameter *type*, used to generate testcases. To get OK for the test, you should print them correctly for at least 90 testcases.

Examples

standard input	standard output
2 3 0 1 3 0 0	-2 1