Problem A. Planting Trees

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Alice and Ivan, having realized that owning a GPU farm is probably not the best for the environment, have decided to make up for their environmental transgressions by planting trees!

As Alice and Ivan are CS majors, they will not plant any old trees, but instead special unrooted trees with N nodes labelled $1, 2, \ldots, N$, such that the degree of every node is an element of the set $D = \{d_1, d_2, \ldots, d_M\}$. Being very creative, they name these D-trees.

Alice and Ivan then plant a single mystery seed, which will sprout into any of the D-trees with equal probability. Being inquisitive, but also no longer having a GPU farm at their disposal, they task you to answer the following: for each $i = 1, 2, \ldots, M$, what is the expected number of nodes in their tree with degree d_i?

Input

The first line of input will have two space-separated integers, N ($2 \le N \le 100\,000$) and M ($1 \le M < N$).

The second line of input will have M space-separated integers, $1 = d_1 < d_2 < \cdots < d_M < N$.

Output

Output M space-separated integers on a single line, the ith of which should equal $p_i q_i^{-1} \pmod{998244353}$, where p_i/q_i is the expected number of nodes with degree d_i, as a fraction in lowest terms, i.e. $\gcd(p_i, q_i) = 1$.

It is guaranteed that the total number of D-trees is not a **non-zero** multiple of 998244353.

Examples

standard input	standard output
3 2 1 2	2 1
6 2 1 3	4 2
5 2 1 3	0 0

Note

For the first example, there is a unique tree, up to relabeling, on 3 vertices whose nodes have degree 2. This is the tree with 2 nodes of degree 1, and 1 node of degree 2.

For the third example, there are no D-trees for $D = \{1, 3\}$. Thus the expected number of nodes with degrees 1 and 3 is 0.

Problem B. Palindromic Triplets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

There is a collection of non-empty strings S that are indexed from 1 to N (where N is the number of strings). Count how many ordered triples of indices (i, j, k) (with repetition allowed) produce a palindrome of odd length when concatenated as $S_i + S_j + S_k$.

Input

The first line of input contains an integer N, the number of strings in S. The following N lines of input consist of 1 or more lowercase letters each, where the i-th line contains the string S_i (string indexed at i). The total number of letters in all of the strings put together will be at most 100,000.

Output

Output a single integer, the number of such triplets.

Examples

standard input	standard output
4 a ab ba cc	6
2 a a	8

Note

So the valid triplets are $(1, 1, 1), (1, 3, 3), (2, 1, 3), (2, 2, 1), (3, 1, 2), (4, 1, 4)$.

So the valid triplets are $(1, 1, 1), (1, 1, 2), (1, 2, 1), (1, 2, 2), (2, 1, 1), (2, 1, 2), (2, 2, 1), (2, 2, 2)$.

Problem C. Magic Matrix

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Generative AI models can do many impressive things. They almost seem like magic, even though underneath, they are just large collections of weights–just numbers. Some AI models can even solve programming contest problems, but please don't use them in a contest because that would be cheating.

Your task here is to construct a matrix with some interesting magic properties. You will be given two sequences of numbers of the same length, for example `4 6 5` and `7 8 9`. You then need to construct a matrix of integers from 1 to 3 (inclusive) of size $4+6+5$ by $7+8+9$ (so 15 by 24), such that:

- In exactly 4 of the rows, strictly more than half the integers are 1
- In exactly 6 of the rows, strictly more than half of the integers are 2
- In exactly 5 of the rows, strictly more than half of the integers are 3
- In exactly 7 of the columns, strictly more than half the integers are 1
- In exactly 8 of the columns, strictly more than half the integers are 2
- In exactly 9 of the columns, strictly more than half the integers are 3

Hopefully, by now you already see how such a matrix can be constructed, as the task at hand is to do so in the general case.

Input

The first line of input contains an integer k, with $1 \leq k \leq 250$, the length of the two given sequences.

The second line contains k integers r_i, with $2 \leq r_i \leq 500$, specifying that in exactly r_i rows of the matrix, strictly more than half of the integers should be i. The sum of all the r_i is at most 500.

The third line contains k integers c_i with $2 \leq c_i \leq 500$, specifying that in exactly c_i columns of the matrix, strictly more than half of the integers should be i. The sum of all the c_i is at most 500.

Output

If the matrix cannot be formed, output a single line containing the word `NO`. If the matrix can be formed, output a line containing the word `YES`, followed by the rows of the matrix, one row per line, with a space separating the integers within a row. If there are multiple possible matrices, output any one of them.

Examples

standard input	standard output
2 2 4 3 2	YES 2 2 1 1 2 2 1 1 1 1 1 1 2 1 1 2 1 2 1 2 2 2 1 2 2 2 1 1 2 2
3 2 2 2 2 2 2	YES 3 2 2 1 2 2 3 1 2 1 1 1 2 1 2 2 3 2 1 1 1 1 3 2 3 3 2 1 3 3 3 1 3 3 3 2

Problem D. The Worm in the Apple

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Willy the Worm was living happily in an apple – until some vile human picked the apple, and started to eat it! Now, Willy must escape! Given a description of the apple (defined as a convex shape in 3D space), and a list of possible positions in the apple for Willy (defined as 3D points), determine the minimum distance Willy must travel to get to the surface of the apple from each point.

Input

There will be several test cases in the input. Each test case will begin with a line with a single integer n ($4 \le n \le 1,000$), which tells the number of points describing the apple.

On the next n lines will be three integers x, y and z ($-10,000 \le x, y, z \le 10,000$), where each point (x, y, z) is either on the surface of, or within, the apple. The apple is the convex hull of these points. No four points will be coplanar.

Following the description of the apple, there will be a line with a single integer q ($1 \le q \le 100,000$), which is the number of queries – that is, the number of points where Willy might be inside the apple. Each of the following q lines will contain three integers x, y and z ($-10,000 \le x, y, z \le 10,000$), representing a point (x, y, z) where Willy might be. All of Willy's points are guaranteed to be inside the apple. The input will end with a line with a single 0.

Output

For each query, output a single floating point number, indicating the minimum distance Willy must travel to exit the apple. Output this number with exactly 4 decimal places of accuracy, using standard 5 up / 4 down rounding (e.g. 2.12344 rounds to 2.1234, 2.12345 rounds to 2.1235). Output each number on its own line, with no spaces, and do not print any blank lines between answers.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>6
0 0 0
100 0 0
0 100 0
0 0 100
20 20 20
30 20 10
4
1 1 1
30 30 35
7 8 9
90 2 2
0</pre></td><td><pre>1.0000
2.8868
7.0000
2.0000</pre></td></tr>
</table>

Problem E. Efficient Room Scheduling

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The university is facing a difficult financial situation and is looking for ways to save as much money as possible. One way to help save money is to pack lectures into rooms that are as small as possible. A smaller room uses less heat and light than a larger room. Unused rooms can be left unheated with the lights turned off.

That said, each room has a maximum capacity that is strictly enforced. Each lecture must be scheduled in a room with at least as much capacity as the number of students in the lecture.

What is the most efficient way to allocate lectures to rooms, to minimize the total capacity of the rooms that are used? Rooms that are not used for any lecture do not count toward the total capacity.

Assume that all the lectures happen at the same time, so that at most one lecture can be allocated to a given room.

Input

The input contains several test cases, at most 100. The first line of each test case contains two integers between 1 and 20000 inclusive, indicating the number n of lectures to be scheduled and the number m of rooms in the university. The next n lines each contain an integer, the number of students in each lecture. The following m lines each contain an integer, the capacity of each room.

The last test case is followed by a line containing `0 0`.

Output

For each test case, output a line containing the minimum total capacity of rooms that need to be allocated to lectures. If it is not possible to fit all the lectures in the rooms, output the a line with the word `Impossible`.

Example

standard input	standard output
2 3 5 4 7 8 4 2 1 5 5 10 0 0	11 Impossible