A. Smallest number
Time limit: 2 seconds

Recently, Vladimir got bad mark in algebra again. To avoid such unpleasant events in future he decided to train his arithmetic skills. He wrote four integers number *a*, *b*, *c*, *d* on the blackboard. During each of the next three minutes he took two numbers from the blackboard (not necessarily adjacent) and replaced them with their sum or their product. In the end he got one number. Unfortunately, due to the awful memory he forgot that number, but he remembers four original numbers, sequence of the operations and his surprise because of the very small result. Help Vladimir remember the forgotten number: find the smallest number that can be obtained from the original numbers by the given sequence of operations.

Input

First line contains four integers separated by space: $0 \le a, b, c, d \le 1000$ — the original numbers. Second line contains three signs ('+' or '*' each) separated by space — the sequence of the operations in the order of performing. ('+' stands for addition, '*' — multiplication)

Output

Output one integer number — the minimal result which can be obtained.

Please, do not use %lld specificator to read or write 64-bit integers in C++. It is preferred to use cin (also you may use %I64d).

Examples

Input
1 1 1 1
+ + *
Output
3
Input
2 2 2 2
* * +
Output
8
Input
1 2 3 4
* + +
Output
9

B. Beautiful numbers
Time limit: 4 seconds

Volodya is an odd boy and his taste is strange as well. It seems to him that a positive integer number is *beautiful* if and only if it is divisible by each of its nonzero digits. We will not argue with this and just count the quantity of beautiful numbers in given ranges.

Input

The first line of the input contains the number of cases t ($1 \leq t \leq 10$). Each of the next t lines contains two natural numbers li and ri ($1 \leq li \leq ri \leq 9 \cdot 1018$).

Please, do not use %lld specificator to read or write 64-bit integers in C++. It is preffered to use cin (also you may use %I64d).

Output

Output should contain t numbers — answers to the queries, one number per line — quantities of beautiful numbers in given intervals (from li to ri, inclusively).

Input

1
1 9

Output

9

Input

1
12 15

Output

2

C. Candy Boxes
Time limit : 1 second

There is an old tradition of keeping 4 boxes of candies in the house in Cyberland. The numbers of candies are *special* if their *arithmetic mean*, their *median* and their *range* are all equal. By definition, for a set $\{x1, x2, x3, x4\}$ $(x1 \le x2 \le x3 \le x4)$ *arithmetic mean* is $\frac{x_1+x_2+x_3+x_4}{4}$, *median* is $\frac{x_2+x_3}{2}$ and *range* is $x4 - x1$. **The arithmetic mean and median are not necessary integer.** It is well-known that if those three numbers are same, boxes will create a "debugging field" and codes in the field will have no bugs.

For example, 1, 1, 3, 3 is the example of 4 numbers meeting the condition because their mean, median and range are all equal to 2.

Jeff has 4 special boxes of candies. However, something bad has happened! Some of the boxes could have been lost and now there are only n $(0 \le n \le 4)$ boxes remaining. The i-th remaining box contains ai candies.

Now Jeff wants to know: is there a possible way to find the number of candies of the $4 - n$ missing boxes, meeting the condition above (the mean, median and range are equal)?

Input

The first line of input contains an only integer n $(0 \le n \le 4)$.

The next n lines contain integers ai, denoting the number of candies in the i-th box $(1 \le a_i \le 500)$.

Output

In the first output line, print "YES" if a solution exists, or print "NO" if there is no solution.

If a solution exists, you should output $4 - n$ more lines, each line containing an integer b, denoting the number of candies in a missing box.

All your numbers b must satisfy inequality $1 \le b \le 106$. It is guaranteed that if there exists a positive integer solution, you can always find such b's meeting the condition. If there are multiple answers, you are allowed to print any of them.

Given numbers a_i may follow in any order in the input, not necessary in non-decreasing.

a_i may have stood at any positions in the original set, not necessary on lowest n first positions.

Examples

Input
2
1
1
Output
YES
3
3
Input
3
1
1
1
Output
NO
Input
4
1
2
2
3
Output
YES
Note
For the first sample, the numbers of candies in 4 boxes can be 1, 1, 3, 3. The arithmetic mean, the median and the range of them are all 2.
For the second sample, it's impossible to find the missing number of candies.
In the third example no box has been lost and numbers satisfy the condition.
You may output *b* in any order

J. ABCDEF

You are given a set S of integers between -30000 and 30000 (inclusive).

Find the total number of sextuples $(a,b,c,d,e,f): a,b,c,d,e,f \in S; d \neq 0$ that satisfy:

$$\frac{a*b+c}{d} - e = f$$

Input

The first line contains integer N ($1 \leq N \leq 100$), the size of a set S.

Elements of S are given in the next N lines, one integer per line. Given numbers will be distinct.

Output

Output the total number of plausible sextuples.

Examples

Input:	**Input:**	**Input:**	**Input:**
1	2	2	3
1	2	-1	5
	3	1	7
			10
Output:	**Output:**	**Output:**	**Output:**
1	4	24	10