Problem A. Black-White Numbers

Input file:	`black.in`
Output file:	`black.out`
Time limit:	0.25 seconds

You are given white integer numbers W_i $(1 \le i \le n)$ and black integer numbers B_i $(1 \le i \le n)$. Write a program which finds such $2n$ positive integers k_i and l_i $(1 \le i \le n)$ that $k_i \ne k_j \forall i \ne j$, $l_i \ne l_j \forall i \ne j$, $1 \le k_i, l_i \le n$, $A = \sum_{1 \le i \le n} |W_{k_i} - B_{l_i}|$ will be minimal possible.

Input

There is a single integer n $(1 \le n \le 10000)$ on the first line. The second line contains n integers — white numbers. The third line contains n integer black numbers. All numbers do not exceed $2,000,000,000$ by absolute value.

Output

Write to the output $n + 1$ lines. In the first line you are to write out minimal possible value of A. In the next n lines write out $2n$ positive integers: $(i + 1)$-th line must contain k_i and l_i, separated by a single space.

Example

black.in	black.out
1 1 2	1 1 1

Problem B. Divide et Impere

Input file:	`divide.in`
Output file:	`divide.out`
Time limit:	0.25 seconds

There is a set of cells Q, which completely lies in a square of $N \times N$ cells. You are to divide Q into two subsets Q_1 and Q_2 in such way, that one could be transformed to another by applying parallel shiftings and rotatings per 90^o.

Input

First line of the input contains single integer number N $(1 \leq N \leq 20)$. Next N lines contains N characters each: '1', if the cell belongs to Q, and '0' in another case.

Output

First line of output file must contain the word "`YES`", if Q could be divided and "`NO`" in another case. In the case of positive answer print one subset Q_1 or Q_2 in N lines, each of which contains N characters: '1', if cell belongs to your subset, and '0' in another case.

Example

`divide.in`	`divide.out`
3 100 011 010	YES 100 010 000
3 100 101 010	NO

Problem C. Creativity

Input file:	`creativ.in`
Output file:	`creativ.out`
Time limit:	0.25 seconds

Famous Saratov artist Madzinski-Kalevich decided to make us happy with his new masterpiece. He has drawn N arcs of circles on a plain sheet of paper. You are to find number of common points of all arcs.

Radius of each circle does not exceed 1000.

Input

First line of the input contains single integer number N ($1 \le N \le 50$). Each of the next N lines contains description of one arc. Description of an arc consists of cartesian coordinates (x, y) of three points: two ends of the arc and an arbitrary point on it, which is not equal to arcs' ends. You may assume that all coordinates don't exceed 1000 by absolute value. Numbers in a line are separated by spaces.

Output

First line of output file must contain M — quantity of different common arcs' points. Next M lines must contain coordinates of these points. Output (x, y) pairs in lexicographical order with 3 decimal digits. Separate x and y by a single space. Here we assume that lexicographically ordered points are sorted by nondescending of first coordinate (x), and by second (y) in the case of a tie. If there are infinitely many common points, output file must contain single word "`Infinity`" without quotes. It is known that if quantity of common points is not infinite, the distance between any two different points is at least 0.005.

Example

creativ.in	creativ.out

3
7 4 7 -4 3 0
2 2 6 2 4 0
0 4 0 -4 4 0
	4
	3.009 0.263
	3.500 -1.936
	3.500 1.936
	4.000 0.000

Problem D. Strange Things

Input file:	`strange.in`
Output file:	`strange.out`
Time limit:	1 second

Long time ago one famous scientist constructed the strange function and the strange sequence. He took some integer positive number N and wrote the following sequence: $A_1 = F(2N, 1)$, $A_2 = F(2N, 3)$, $A_3 = F(2N, 5) \ldots A_N = F(2N, 2N - 1)$ where $F(P, Q)$ is a strange function, which was constructed by the following rules:

1. For each integer positive number $Q : F(0, Q) = 0$
2. For each integer non-negative number $P : F(P, 0) = 1$
3. Otherwise $F(P, Q) = F(P - 1, Q - 1) + F(P - 1, Q)$

But the primary goal of his research was the strange function $X(N)$. He denoted by $X(N)$ the greatest common divisor of the numbers $A_1 \ldots A_N$. He finished exploring this problem after some years of unsuccessful attempts. So, as you understand, your task is quite simple. You have to calculate function $X(N)$ for a given positive integer N.

Input

Input file contains one integer number N ($1 \le N \le 10^{10000}$).

Output

You must output $X(N)$.

Example

strange.in	strange.out
1	2

Problem E. $A + B = C$

Input file:	`abc.in`
Output file:	`abc.out`
Time limit:	0.25 seconds

You are given an array $A[1 \dots N]$ of different integer numbers (all of its elements do not exceed $15 \cdot 10^6$ by their absolute value). Your task is to calculate the number of different triples (p, q, r), such that $A[p] + A[q] = A[r]$ and p, q and r are different. Note that triples $(1, 2, 3)$ and $(2, 1, 3)$ are considered to be different.

Input

The first line of input file contains positive integer N $(0 < N < 3501)$. The next line consists of the numbers $A[1], A[2], \dots, A[N]$ separated by spaces.

Output

Write the only number — answer to this problem.

Example

abc.in	abc.out
4 3 2 4 1	4

Problem F. Simple Calculations

Input file:	`calc.in`
Output file:	`calc.out`
Time limit:	0.25 seconds

You have to calculate the function $F(n, k) = \left(2^{2^n} + 1\right) \bmod k$.

Input

Input file contains two integer numbers n and k separated by spaces ($0 \le n \le 2^{31} - 1$, $1 \le k \le 10^6$).

Output

You must output $F(n, k)$.

Example

calc.in	calc.out
0 4	3

Problem G. The Game (Building a Palindrome)

Input file:	`game.in`
Output file:	`game.out`
Time limit:	0.25 seconds

Two men are playing the following game. The game lasts $2n + 1$ rounds. In each round players make their moves. The first player chooses a letter (either 'A' or 'B') which is to be appended to the right end of the current word. Then the second player may swap two letters in the resulting word (which is now one letter longer than before). He can make at most one swap per turn (thus he can do nothing). Initially the word is empty. The second player wins if the final word (the word after the game is over) is a palindrome (palindrome is a sequence of letters which doesn't change after reversion). Your task is to write a program that plays for the second player in the best way (i. e. it gets a palindrome whenever it's possible).

Input

There are up to five test cases in the input file. In the first line there is a number of games K ($1 \le K \le 5$). The following lines contain K test cases. Each test case consists of two lines. The first line of the test case contains an integer number n ($0 \le n \le 127$). In the second line there is a sequence of $2n + 1$ symbols 'A' and 'B' without spaces, which is the sequence of moves of the first player.

Output

Write out K groups of lines. The number of lines in the i-th group must be equal to the number of rounds in the i-th game (i. e. in the i-th test case). In the j-th line of the i-th group write out two numbers separated by a single space — the positions of letters which should be swapped by the second player in the j-th round or write out '`0 0`' if no swap should occur. If the second player can't win, the whole group must consist of lines '`-1 -1`'.

Note

Your program should work in such way that it only knows first j moves of the first player when choosing a move for j-th round. (i. e. for any two games in which first X moves of the first player match, the first X moves of the second player must also match)

Example

`game.in`	`game.out`
`2`	`0 0`
`1`	`1 2`
`BAA`	`0 0`
`3`	`0 0`
`AAABAAA`	`0 0`
	`0 0`
	`0 0`
	`0 0`
	`0 0`
	`0 0`

Problem H. Sieve Coding

Input file:	`sieve.in`
Output file:	`sieve.out`
Time limit:	0.25 seconds

The coding process known as “Sieve’ is made in the following way. The letters of the text are being written in cells of $N \times N$ matrix consequently from left to right, from top to bottom (the rest of the matrix is being filled with ‘*’ symbol). The key of the code is a card (sieve) with the grid marked on it (the grid has the same size i. e. number of rows and columns as the matrix) which has some cells cut out from it. This card is laid upon the matrix and the letters are written out from the visible cells of the matrix in the order they appear. Then the sieve is turned 90 degrees clockwise and visible letters are written out again. Such an operation repeats 4 times. You should write a program that answers whether it is possible to encrypt a given string using a given sieve (it is considered that it can be done if each cell of the matrix occurs in encrypted string exactly once) and outputs the encrypted string.

Input

There is a non-empty string S in the first line of the input file (it consists of the uppercase Latin letters and its length doesn’t exceed 256). The second line contains integer number N $(0 < N < 17)$. Next N lines contain N symbols each. These N lines represent the description of the sieve. Symbol ‘`+`’ represents a hole in a corresponding cell of the grid and symbol ‘`.`’ (dot) represents an opaque cell of the grid.

Output

Write ‘`No solution.`’ in the first line of the output file if the string can’t be encrypted and write the result of encryption otherwise.

Example

sieve.in	sieve.out

`ABCD`
`2`
`.+`
`..`
	`BDCA`

Problem I. Ellipse and the Line

Input file:	`ellipse.in`
Output file:	`ellipse.out`
Time limit:	0.25 seconds

You are given an ellipse $x^2/a^2 + y^2/b^2 = 1$ by real numbers a, b ($1 \le a, b \le 100$) and a straight line that goes through points (x_1, y_1) and (x_2, y_2) with real coordinates ($-500 \le x_1, y_1, x_2, y_2 \le 500$). It is required to find the ratio of the smaller area to the greater area of parts of an ellipse into which it is broken with a straight line. If the straight line does not cross an ellipse, the given ratio should be considered equal to zero.

Input

Input file contains numbers a, b, x_1, y_1, x_2, y_2 ($(x_1, y_1) \ne (x_2, y_2)$). All numbers are real and consist of no more than two digits after decimal point.

Output

You must output the required ratio with precision of six digits after decimal point.

Example

`ellipse.in`	`ellipse.out`

`100 100`
`1 0`
`4 2`
	`0.985974`

Problem J. Sea battle

Input file:	`sea.in`
Output file:	`sea.out`
Time limit:	0.25 seconds

You are given locations of ships on a 10×10 field in "Sea battle" game. According to game rules, field must contains the following ships:

- one four-decks;
- two three-decks;
- three double-decks;
- four single-decks.

Ships should be straight, oriented horizontally or vertically and can not touch each other. You are to write a program, that checks whether the set of play fields matchs described rules.

Input

First line of the input contains single integer number N ($1 \leq N \leq 10$) — quantity of play fields. N descriptions of play fields follows. Each description contains 10 lines with 10 characters in each. Symbol '0' means that the corresponding cell is free, and '*' means that corresponding cell is covered by a ship. Fields descriptions are separated by blank lines.

Output

Output file must consist of N lines. Each line should contain the word '`YES`' if the corresponding field is correct and '`NO`' otherwise.

Example

sea.in	sea.out
1 ****000000 0000000000 ***00***00 0000000000 00000000** 000**00000 00000000** 000*000000 00000*00*0 0*00000000	YES