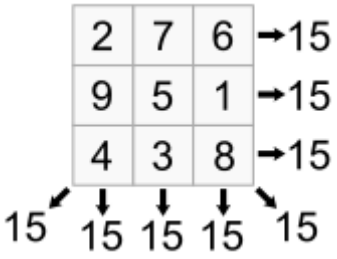
Problem P1 Magic Squares

A magic square is a two-dimensional arrangement of positive integers with a special property: All rows, columns and diagonals total the same number. Below, for example, the 3-by-3 arrangement of 9 numbers is a magic square:



Your assignment is to read a square of numbers as input and determine whether it is a magic square or not. Your only concern is that the totals are all the same: in particular, you **do not** need to check whether a number is duplicated; you **do not** need to check whether the square uses the numbers 1 through N^2.

Input Format

Your program will read from standard input. The first line will contain a positive integer $2 < N < 6$ which represents the number of columns and rows in the square. The next N lines will each contain N positive integers separated by spaces.

Output Format

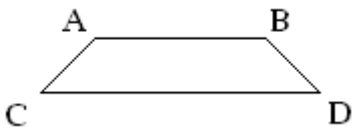
Your program will write to standard output. Your program should produce a single line of output on a line by itself. The line will contain either MAGIC or NO.

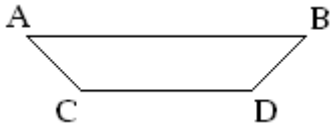
Sample Input and Corresponding Sample Output

Sample Input	Sample Output
3 2 7 6 9 5 1 4 3 8	MAGIC
3 1 2 3 4 5 6 7 8 9	NO
4 7 12 1 14 2 13 8 11 16 3 10 5 9 6 15 4	MAGIC

Problem P2 Trapezoid Rules

A trapezoid is a quadrilateral defined by four points (A, B, C, and D) with two parallel sides defined by the lines AB and CD. In an isosceles trapezoid (shown below) the lines AC and BD are at ±45 degrees.





Your task is to output to the console a trapezoid given two integer values, X=length of AB and Y= length of CD.

Input Format

Your program will read from standard input a single line containing two integers X and Y on a line by themselves separated by a space. X and Y will both be **even** integers. You can assume that $|X - Y| \geq 4$, $2 \leq X \leq 20$ and $2 \leq Y \leq 20$.

Output Format

Your program will write to standard output. When you output the trapezoid to the console, you must ensure that the trapezoid is flush left with the output. To make it easy to validate your output, you will actually output the smallest rectangular region that contains the trapezoid. You must use an asterisk ("*") for the body of the trapezoid and a period (".") for the unused space.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
4 10	...****... ..******.. .********. **********
8 4	******** .******. ..****..
2 6	..**.. .****. ******

Problem P3 Pangramania

A *pangram* is a phrase that uses every letter of the English alphabet at least once. One of the more popular pangrams in the English language is:

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Your task is to determine whether a given phrase is a pangram.

Input Format

Your program will read from standard input a single line containing at least 26 but no more than 80 characters. This line of input will only contain spaces and uppercase characters and will be terminated by the end of line character.

Output Format

Your program will write to standard output a single phrase in uppercase characters on a line by itself. The output will be either PANGRAM or NO.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
CWM FJORD BANK GLYPHS VEXT QUIZ	PANGRAM
ABCD EFGH IJKLM NOPQ RSTU VWXY ABCD	NO
THE FIVE BOXING WIZARDS JUMP QUICKLY	PANGRAM

Problem P4 Roll the Dice

You are a dice maker and want to create a pair of special six-sided dice. You can put any number between 1 and 9 on each side of a die (the same number may be appear more than once on the same die). You want to know the exact probability to roll any value between 2 and 18. You are not interested in values that cannot possibly occur.

Input Format

Your program will read from standard input. The input will consist of two lines of input. Each line will contain six numbers separated by a space, representing the values that appear on the six sides of a die.

Output Format

Your program will write to standard output the table of probabilities, in increasing order of the value of the roll of the two dice. Each line of output contains two parts separated by a single space; the first part is "v." (i.e., the value followed by a period character) and the second part is "p/36" where p is the probability of the value v occurring given the dice. You must omit from the output any values (between 2 and 18) that have zero chance of occurring, given the dice structure

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
1 2 3 4 5 6 6 5 4 3 2 1	2. 1/36 3. 2/36 4. 3/36 5. 4/36 6. 5/36 7. 6/36 8. 5/36 9. 4/36 10. 3/36 11. 2/36 12. 1/36
1 3 5 7 9 9 2 4 6 8 8 8	3. 1/36 5. 2/36 7. 3/36 9. 6/36 11. 7/36 13. 6/36 15. 5/36 17. 6/36
1 1 1 2 2 2 3 3 4 4 5 5	4. 6/36 5. 12/36 6. 12/36 7. 6/36

Problem P5 Euler's Gold

Euler's totient function $\varphi(n)$ is defined as the number of positive integers $\leq n$ that are relatively prime to n (i.e., do not share a common factor), where 1 is counted as being relatively prime to all numbers.

Consider $\varphi(15)$. 2 is relatively prime to 15 but 6 is not (since both 6 and 15 have 3 as a common factor). In the following table, 8 entries are relatively prime to 15, so $\varphi(15) = 8$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Relatively Prime?	Yes	Yes		Yes			Yes	Yes			Yes		Yes	Yes	

Note that if n is a prime number, then $\varphi(n) = n-1$ by the very definition of a prime number. Your task is to compute $\varphi(n)$ for any $1 < n < 1024$.

Input Format

Your program will read from standard input. The input will consist of a single integer n on a line by itself. You can assume that $1 < n < 1024$.

Output Format

Your program will write to standard output a single integer on a line by itself which represents $\varphi(n)$.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
15	8
7	6
16	8
14	6
10	4

Problem P6 Word Triangle

You are given as input a sequence of lines where each line has a single word on the line by itself (i.e., there are no spaces and all letters are in upper case). Your job is to determine whether the sequence represents a *word triangle*. The first word of a word triangle consists of a single letter. Each subsequent line contains a word formed by scrambling the letters from the previous word and adding an additional letter. The following sequence of words forms a word triangle:

```
A
AT
SAT
STAY
YEAST
STAYED
STRAYED
```

Note that a triangle starts with a one-letter word and each successive line uses all of the letters from the previous line and adds a new one (thus "STAYED" = "YEAST" + "D"). Your task is to determine if a sequence of words forms a word triangle.

Input Format

Your program will read from standard input. The first line will contain a positive integer $2 < N < 10$ representing the number of lines that follow. Each of the succeeding N lines contains one uppercase word on a line by itself (each line has fewer than 10 characters).

Output Format

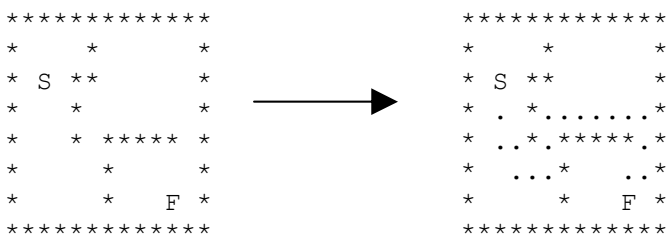
Your program will write to standard output a single phrase in uppercase characters on a line by itself. The output will be either TRIANGLE or NO.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
7 A AT SAT STAY YEAST STAYED STRAYED	TRIANGLE
4 A RA ARA ARRR	NO
7 O TO NOT TONE NOTED DENOTE ONETOED	TRIANGLE

Problem P7 Amazing

You are given a rectangular field, bounded on the outside edge by an impenetrable wall. In this field there is a designated **Start** position where a robot is placed. There is another designated **Finish** position where the robot must end up. The robot can move to any of its four neighboring squares (UP, DOWN, LEFT or RIGHT) provided that square does not contain a wall. The goal is to identify the length of the **shortest** path from Start to Finish (you can assume that the path always exists). A wall is represented by the "*" character, the start position by an "S", the finish position by an "F" and an empty spot is represented by a space (" "). In the maze below, the robot needs 18 steps to get from "S" to "F":



Input Format

Your program will read from standard input. The first line will contain two positive integer W and H separated by a space, where $3 \le W \le 15$ and $3 \le H \le 15$. The remaining H lines of input each contain W characters and represent the maze using character "*" to represent a wall, the space (" ") character is an empty room, the "S" character is the start position and the "F" character is the end position. There will only be a single "S" and "F" character in the input and you can be assured that there is indeed a path (moving only UP, DOWN, LEFT and RIGHT) from the Start position to the Finish position.

Output Format

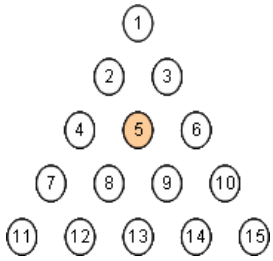
Your program will write to standard output a single integer on a line by itself to represent the number of steps it takes the robot to go from Start to Finish.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
13 8 ************* * * * * S ** * * * * * * ***** * * * * * * F * *************	18
3 4 *** *S* *F* ***	1

Sample Input	Sample Output
7 5 ******* *S * * * * * F* *******	6
9 3 ********* * S F * *********	2

Problem P8 I've got you pegged



The solitaire peg game contains fifteen holes in a triangular arrangement that is numbered as shown on the right. Initially, fourteen pegs are placed in all holes except for the one labeled "5". To move a peg in hole X, you must jump over another peg in hole Y (like in checkers) and place the original peg into an empty hole Z. This move is labeled "X Z". The peg in hole Y is removed from the board. You can only jump diagonally or horizontally and you must end up in an empty hole.

For example, in the initial board the move "12 5" is valid but the move "11 4" is not valid (since there is a peg in the "4" hole). The move "1 7" is not valid because the holes are too far apart to allow a jump from "1" to "7".

You are given a sequence of valid moves and your job is to output the location (in order) of the remaining pegs on the board.

Input Format

You receive as input an integer N>0 on a line by itself describing the number of moves that were made in the game. On each of the subsequent N lines you will find two positive integers X and Z separated by a space representing the move of a peg from hole "X" to hole "Z". You can be assured that all moves are valid moves.

Output Format

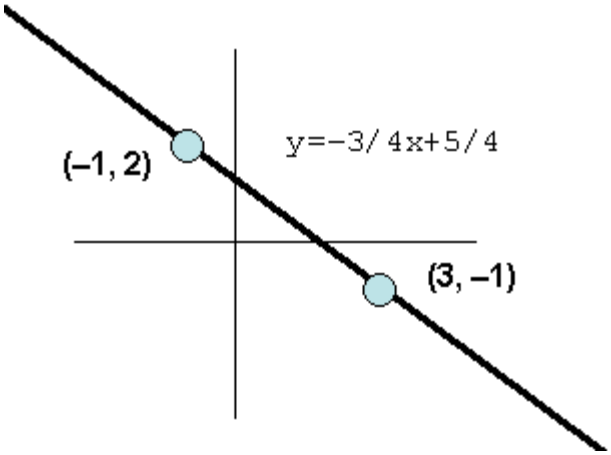
The output will consist of a sequence of lines, each containing a single integer on that line by itself representing a hole that contains a peg. You must have as many lines of output as there are pegs left in the board. The integers must appear in increasing sorted order.

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
10 12 5 14 12 11 13 4 11 5 14 14 12 11 13 1 4 6 1 15 6	1 4 6 13
4 12 5 3 8 14 5 8 3	1 2 3 4 6 7 10 11 13 15

Problem P9 It's not the height… it's the slope

It is well known that two points (x_1, y_1) and (x_2, y_2) in the Cartesian plane describe a straight line. All non-vertical lines can be described using the equation $y=mx+b$ where m is the slope of the line and b is the y-intercept (the point on the y axis that intersects the line). Your task is to output the equation representing the line.



Input Format

You receive as input four integers on a single line separated by spaces. These four integers represent the values x_1, y_1, x_2, y_2. You can assume that the point (x_1, y_1) is not identical to the point (x_2, y_2). You can assume that $x_1 \neq x_2$ (i.e., no vertical lines). All of the values x_1, y_1, x_2, y_2 will be greater than –128 and lesser than 128.

Output Format

The output will consist of a single line containing the equation $y=mx+b$ with no spaces. The goal is to output as few characters as possible. To do this, you must conform to the following constraints:

C1. Should m be 1, omit its value and output $y=x+b$; similarly, should m be –1, omit its value and output $y=–x+b$
C2. Should m be a fraction, output $y=u/vx+b$ where u/v is the most reduced form of m
C3. Should m be a whole integer, output $y=mx+b$
C4. Should m be 0, omit its value and "x" and output $y=b$
C5. Should b be 0, omit its value and output $y=mx$
C6. Should b be negative, then output $y=mx–b$
C7. Should b be a fraction, output $y=mx+u/v$ where u/v is the most reduced form of b
C8. Should b be a whole integer, output $y=mx+b$

Sample Input and Corresponding Sample Output

Sample Input	Sample Output
10 10 50 50	y=x
-10 10 10 -10	y=-x
1 2 3 4	y=x+1
10 -30 20 -30	y=-30
4 2 7 1	y=-1/3x+10/3
0 0 2 4	y=2x
-4 -2 -7 -1	y=-1/3x-10/3