Problem A. Tic Tac Toe

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

The game of Tic Tac Toe is played on an $n \times n$ grid (where n is usually but not necessarily three). Two players alternate placing symbols on squares of the grid. One player places X's and the other player places O's. The player placing X's always goes first. When the grid contains a vertical, horizontal, or diagonal sequence of at least m consecutive squares all containing the same symbol, the game ends and the winner is the player who placed the last symbol. When all the squares of the grid are filled, if neither player has won, the game ends in a draw. Your task is to analyze the state of a Tic Tac Toe board, and determine whether the game is still in progress, or if it has completed, who won, or if the game ended in a draw. You should also detect erroneous states of the Tic Tac Toe board that could never occur during an actual game.

Input

The first line of input contains the two integers n and m, separated by spaces, with $1 \le m \le n \le 2000$. The following n lines of input each contain one row of the Tic Tac Toe board. Each of these lines contains exactly n characters, and each of these characters is either an '`X`', an '`O`', or a period ('`.`'), indicating an empty square.

Output

Output a single line containing the appropriate string "`X WINS`", "`O WINS`", or "`DRAW`" if the game is over, the string "`IN PROGRESS`" if the game has not yet finished, or "`ERROR`" if the state of the board could never occur during a game.

Example

Standard input	Standard output
3 3 ..X OOX ..X	X WINS

Problem B. Nice Prefixes

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

Consider strings formed from characters from an alphabet of size K. For example, if $K = 4$, our alphabet might be $\{a, b, c, d\}$, and an example string is "`bbcac`". For a string S, define $count(S, k)$ to be the number of occurrences of the symbol k in S. For example, $count(bbcac, b) = 2$ and $count(bbcac, a) = 1$.

A prefix of a string S is any string obtained from S by deleting some (possibly none) of the trailing characters of S. For example, the prefixes of "`acb`" are the empty string, "`a`", "`ac`, and "`acb`".

A string S has *nice prefixes* if for every prefix P of S and for every two characters k_1 and k_2 in the alphabet, $|count(P, k1) - count(P, k2)| \leq 2$. For example, "`bbcac`" has nice prefixes, but "`abbbc`" does not because $count(abbb, b) = 3$ and $count(abbb, c) = 0$.

Count the number of strings of length L on an alphabet of size K that have nice prefixes. This number can be large, so print its remainder when divided by $10^9 + 7$.

Input

The input is a single line containing the two integers L and K, separated by spaces, with $1 \leq L \leq 10^{18}$ and $1 \leq K \leq 50$.

Output

Output a single line containing the number of strings of length L on an alphabet of size K that have nice prefixes, modulo $10^9 + 7$.

Example

Standard input	Standard output
`4 2`	`12`

Problem C. Slalom

Input file: `Standard input`
Output file: `Standard output`
Memory limit: 256 mebibytes

You are competing in a ski slalom, and you need to select the best skis for the race. The format of the race is that there are N pairs of left and right gates, where each right gate is W metres to the right of its corresponding left gate, and you may neither pass to the left of the left gate nor to the right of the right gate. The ith pair of gates occurs at distance yi down the hill, with the horizontal position of the i-th left gate given by x_i. Each gate is further down the hill than the previous gate (i.e. $y_i < y_{i+1}$ for all i).

You may select from S pairs of skis, where the jth pair has speed s_j. Your movement is governed by the following rule: if you select a pair of skis with speed s_j, you move with a constant downward velocity of s_j metres per second. Additionally, at any time you may move at a horizontal speed of at most v_h metres per second.

You may start and finish at any two horizontal positions. Determine which pair of skis will allow you to get through the race course, passing through all the gates, in the shortest amount of time.

Input

The first line of input contains the three integers W, v_h, and N, separated by spaces, with $1 \le W \le 10^8$, $1 \le v_h \le 10^6$, and $1 \le N \le 10^5$. The following N lines of input each contain two integers x_i and y_i, the horizontal and vertical positions respectively of the ith left gate, with $1 \le x_i, y_i \le 10^8$.

The next line of input contains an integer S, the number of skis, with $1 \le S \le 10^6$.

The following S lines of input each contain one integer s_j, the speed of the j-th pair of skis, with $1 \le sj \le 10^6$.

Output

If it is impossible to complete the race with any pair of skis, print the line "`IMPOSSIBLE`". Otherwise, print the vertical speed s_j of the pair of skis that allows you to get through the race course in the shortest time.

Examples

<table>
<tr><th>Standard input</th><th>Standard output</th></tr>
<tr><td><pre>3 2 3
1 1
5 2
1 3
3
3
2
1</pre></td><td><pre>2</pre></td></tr>
<tr><td><pre>3 2 3
1 1
5 2
1 3
1
3</pre></td><td><pre>IMPOSSIBLE</pre></td></tr>
</table>

Problem D. Celebrity Split

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

Jack and Jill have decided to separate and divide their property equally. Each of their N mansions has a value between 10^6 and $4 \cdot 10^7$ dollars. Jack will receive some of the mansions; Jill will receive some of the mansions; the remaining mansions will be sold, and the proceeds split equally.

Neither Jack nor Jill can tolerate the other receiving property with higher total value. The sum of the values of the mansions Jack receives must be equal to the sum of the values of the mansions Jill receives. So long as the value that each receives is equal, Jack and Jill would like each to receive property of the highest possible value.

Given the values of N mansions, compute the value of the mansions that must be sold so that the rest may be divided so as to satisfy Jack and Jill.

Input

The input consists of a sequence of test cases. The first line of each test case contains a single positive integer N, the number of mansions, which will be no more than 24. This line is followed by N lines, each giving the value of a mansion. The final line of input contains the integer zero. This line is not a test case and should not be processed.

Output

For each test case, output a line containing a single integer, the value of the mansions that must be sold so that the rest may be divided so as to satisfy Jack and Jill.

Example

Standard input	Standard output
5 6000000 30000000 3000000 11000000 3000000 0	41000000

Note

Suppose Jack and Jill own 5 mansions valued at $6 \cdot 10^6$, $3 \cdot 10^7$, $3 \cdot 10^6$, $1.1 \cdot 10^7$, and $3 \cdot 10^6$ dollars. To satisfy their requirements, Jack or Jill would receive the mansion worth $6 \cdot 10^6$ and the other would receive both manstions worth $3 \cdot 10^6$ dollars. The other mansions would be sold, for a total of $4.1 \cdot 10^7$. The answer is therefore 41000000.

Problem E. Knight's Trip

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

In chess, each move of a knight consists of moving by two squares horizontally and one square vertically, or by one square horizontally and two squares vertically. A knight making one move from location $(0,0)$ of an infinite chess board would end up at one of the following eight locations: $(1,2)$, $(-1,2)$, $(1,-2)$, $(-1,-2)$, $(2,1)$, $(-2,1)$, $(2,-1)$, $(-2,-1)$.

Starting from location $(0,0)$, what is the minimum number of moves required for a knight to get to some other arbitrary location (x,y)?

Input

Each line of input contains two integers x and y, each with absolute value at most 10^9. The integers designate a location (x,y) on the infinite chess board. The final line contains the word "`END`".

Output

For each location in the input, output a line containing one integer, the minimum number of moves required for a knight to move from $(0,0)$ to (x,y).

Example

Standard input	Standard output
1 2 2 4 END	1 2

Problem F. Paintball

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

You are playing paintball on a 1000×1000 square field. A number of your opponents are on the field hiding behind trees at various positions. Each opponent can fire a paintball a certain distance in any direction. Can you cross the field without being hit by a paintball?

You must enter the field somewhere between the southwest and northwest corner and must leave somewhere between the southeast and northeast corners.

Input

Assume that the southwest corner of the field is at $(0, 0)$ and the northwest corner at $(0, 1000)$. The input consists of a line containing $n \leq 1000$, the number of opponents. A line follows for each opponent, containing three real numbers: the (x, y) location of the opponent and its firing range. The opponent can hit you with a paintball if you ever pass within his firing range.

Output

If you can complete the trip, output four real numbers with two digits after the decimal place, the coordinates at which you may enter and leave the field, separated by spaces. If you can enter and leave at several places, give the most northerly. If there is no such pair of positions, print line, containing one number -1.

Example

Standard input	Standard output

`3`
`500 500 499`
`0 0 999`
`1000 1000 200`
	`0.00 1000.00 1000.00 800.00`

Problem G. Fire!

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

Joe works in a maze. Unfortunately, portions of the maze have caught on fire, and the owner of the maze neglected to create a fire escape plan. Help Joe escape the maze.

Given Joe's location in the maze and which squares of the maze are on fire, you must determine whether Joe can exit the maze before the fire reaches him, and how fast he can do it.

Joe and the fire each move one square per minute, vertically or horizontally (not diagonally). The fire spreads all four directions from each square that is on fire. Joe may exit the maze from any square that borders the edge of the maze. Neither Joe nor the fire may enter a square that is occupied by a wall.

Input

The first line of input contains the two integers R and C, separated by spaces, with $1 \le R, C \le 1000$. The following R lines of input each contain one row of the maze. Each of these lines contains exactly C characters, and each of these characters is one of:

- '`#`', a wall;
- '`.`', a passable square;
- '`J`', Joe's initial position in the maze, which is a passable square;
- '`F`', a square that is on fire

There will be exactly one '`J`' in the input.

Output

Output a single line containing "`IMPOSSIBLE`" if Joe cannot exit the maze before the fire reaches him, or an integer giving the earliest time Joe can safely exit the maze, in minutes.

Examples

<table>
<tr><th>Standard input</th><th>Standard output</th></tr>
<tr><td><pre>4 4
####
#JF#
#..#
#..#</pre></td><td><pre>3</pre></td></tr>
<tr><td><pre>3 3
###
#J.
#.F</pre></td><td><pre>IMPOSSIBLE</pre></td></tr>
</table>

Problem H. Alaska

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

The Alaska Highway runs 1422 miles from Dawson Creek, British Columbia to Delta Junction, Alaska. Brenda would like to be the first person to drive her new electric car the length of the highway. Her car can travel up to 200 miles once charged at a special charging station. There is a charging station in Dawson Creek, where she begins her journey, and also several charging stations along the way. Can Brenda drive her car from Dawson City to Delta Juntion and back?

Input

The input contains several scenario. Each scenario begins with a line containing n, a positive number indicating the number of charging stations. n lines follow, each giving the location of a filling station on the highway, including the one in Dawson City. The location is an integer between 0 and 1422, inclusive, indicating the distance in miles from Dawson Creek. No two filling stations are at the same location. A line containing 0 follows the last scenario.

Output

For each scenario, output a line containing "`POSSIBLE`" if Brenda can make the trip. Otherwise, output a line containing the word "`IMPOSSIBLE`".

Example

Standard input	Standard output
2 0 900 8 1400 1200 1000 800 600 400 200 0 0	IMPOSSIBLE POSSIBLE

Problem I. Driving Range

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

These days, many carmakers are developing cars that run on electricity instead of gasoline. The batteries used in these cars are generally very heavy and expensive, so designers must make an important tradeoffs when determining the battery capacity, and therefore the range, of these vehicles. Your task is to help determine the minimum range necessary so that it is possible for the car to travel between any two cities on the continent.

The road network on the continent consists of cities connected by bidirectional roads of different lengths. Each city contains a charging station. Along a route between two cities, the car may pass through any number of cities, but the distance between each pair of consecutive cities along the route must be no longer than the range of the car. What is the minimum range of the car so that there is a route satisfying this constraint between every pair of cities on the continent?

Input

The input consists of a sequence of road networks. The first line of each road network contains two positive integers n and m, the number of cities and roads. Each of these integers is no larger than one million. The cities are numbered from 0 to $n-1$. The first line is followed by m more lines, each describing a road. Each such line contains three non-negative integers. The first two integers are the numbers of the two cities connected by the road. The third integer is the length of the road. The last road network is followed by a line containing two zeros, indicating the end of the input.

Output

For each road network, output a line containing one integer, the minimum range of the car that enables it to drive from every city to every other city. If it is not possible to drive from some city to some other city regardless of the range of the car, instead output a line containing the word "`IMPOSSIBLE`".

Example

Standard input	Standard output
3 3 0 1 3 1 2 4 2 1 5 2 0 0 0	4 IMPOSSIBLE

Problem J. Buzzwords

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

The word "the" is the most common three-letter word. It even shows up inside other words, such as "o**the**r" and "ma**the**matics". Sometimes it hides, split between two words, such as "no**t he**re". Have you ever wondered what the most common words of lengths other than three are?

Your task is the following. You will be given a text. In this text, find the most common word of length one. If there are multiple such words, any one will do. Then count how many times this most common word appears in the text. If it appears more than once, output how many times it appears. Then repeat the process with words of length 2, 3, and so on, until you reach such a length that there is no longer any repeated word of that length in the text.

Input

The input consists of a sequence of lines. The last line of input contains single zero ('`0`') and should not be processed. Each line of input other than the last contains at least one but no more than one thousand uppercase letters and spaces. The spaces are irrelevant and should be ignored.

Output

For each line of input, output a sequence of lines, giving the number of repetitions of words of length 1, 2, 3, and so on. When you reach a length such that there are no repeated words of that length, output number 1.

Example

Standard input	Standard output
OTHER MATHEMATICS NOT HERE AA 0	5 4 4 2 2 1 2 1

Problem K. Ferry Loading

Input file:	`Standard input`
Output file:	`Standard output`
Memory limit:	256 mebibytes

Before bridges were common, ferries were used to transport vehicles across rivers. River ferries, unlike their larger cousins, run on a guide line and are powered by the river's current. Two lanes of vehicles drive onto the ferry from one end, the ferry crosses the river, and the vehicles exit from the other end of the ferry.

The vehicles waiting to board the ferry form a single queue, and the operator directs each vehicle in turn to drive onto the port (left) or starboard (right) lane of the ferry so as to balance the load. Each vehicle in the queue has a different weight, which the operator estimates by inspecting the queue.

Given n vehicles, you are to compute how to divide them between the port and starboard lanes so that the weight of vehicles on each side is nearly equal.

Input

Input contains not more than 400 test cases. Each test case begins with $1 < n \leq 100$; the number of vehicles to be boarded. Consider the vehicles to be numbered 1 through n. n lines follow; the i-th line gives the weight in tonnes of the ith vehicle — a positive real number not greater than 100. A line containing 0 follows the last test case.

Output

For each test case, output a single line giving the numbers of the vehicles that should be directed to the starboard lane. Assume that the other vehicles will be directed to the port lane. The total weight of vehicles on the port side should not differ from that on the starboard side by more than 2%. If there are many solutions, any one will do. There will always be a solution; indeed, there will always be a solution that balances within 1%, but you aren't required to find it.

Example

Standard input	Standard output
5 10.0 50.0 90.0 38.0 7.1 0	3 5

Problem L. Frosh Week

Input file: `Standard input`
Output file: `Standard output`
Memory limit: 256 mebibytes

During Frosh Week, students play various fun games to get to know each other and compete against other teams. In one such game, all the frosh on a team stand in a line, and are then asked to arrange themselves according to some criterion, such as their height, their birth date, or their student number. This rearrangement of the line must be accomplished only by successively swapping pairs of consecutive students. The team that finishes fastest wins. Thus, in order to win, you would like to minimize the number of swaps required.

Input

The first line of input contains one positive integer n, the number of students on the team, which will be no more than one million. The following n lines each contain one integer, the student number of each student on the team. No student number will appear more than once.

Output

Output a line containing the minimum number of swaps required to arrange the students in increasing order by student number.

Example

Standard input	Standard output
3 3 1 2	2