Problem A. A ICPC-strings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider the ICPC-string as the string consisting of letters ICPC such that any 4 consecutive letters contain two letters '`C`', one letter '`I`' and one letter '`P`'.

For given $N \geq 4$ your task is to find the number of the different ICPC-strings of length N.

Input

The input contains one integer N — the length of the string ($4 \leq N \leq 1000$).

Output

Print one integer — the number of ICPC-strings of length N.

Examples

standard input	standard output
5	12

Note

In the sample, we have the following 12 ICPC-strings of the length 5: `ICPCI`, `CCIPC`, `CCPIC`, `CICPC`, `CIPCC`, `CPCIC`, `CPICC`, `ICCPI`, `IPCCI`, `PCCIP`, `PCICP`, `PICCP`.

Problem B. Beautiful Array

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider the array $A[i]$ $(i = 0, 1 \ldots, N-1)$ of length N, consisting of positive integers, **beautiful**, if for any $1 \leq i \leq N$ decimal representations of $A[i]$ and i does not have common digits (for example, if $i = 2022$, then $A[i]$ can be equal 19, but it cannot be equal 303, because the zero can be found in both 303 and 2002).

For given N check if exists the beautiful array of length N.

Input

Input contains one integer N — the length of the array $(1 \leq N \leq 10^{18})$.

Output

Print 1, if exists the beautiful array of length N, and 0 otherwise.

Examples

standard input	standard output
3	1

Note

For the sample test, one of possible diverse arrays is $A[0] = 1$, $A[1] = 2$, $A[2] = 4$.

Problem C. Create The Scaled Picture

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given the picture $n \times m$, built from black and white pixels. Your task is to scale it up, replacing each pixel with the rectangle $a \times b$, entirely consisting of the pixels of same color as the replaced pixel.

Input

The first line of the input contains four integers n, m, a, b ($1 \le n, m, a, b \le 10$).

i-th of the following n lines contain the binary string of length m, describing i-th row of the initial picture. '0' denotes the white pixel; '1' denotes the black pixel.

Output

Print the scaled picture as the rectangle consisting of $n \times a$ rows and $m \times b$ columns.

Examples

standard input	standard output
2 2 1 1 10 11	10 11
2 2 2 2 10 11	1100 1100 1111 1111
2 2 2 3 10 11	111000 111000 111111 111111

Problem D. Distance To Special Point

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

This is an interactive problem.

On the plane the integer points with coordinates does not exceeding 10^6 are labeled. The movement is allowed along the lines that are parallel to the coordinate axis, so the distance between two points that have coordinates (x_1, y_1) and (x_2, y_2) is calculated as $|(x1 - x2)| + |(y1 - y2)|)$.

There is unknown special labeled point A. You may for one query ask the distance from the labeled point you choose to the point A. Your task is to guess the coordinates of A for two queries.

Interaction Protocol

The interaction is started by your program. You may ask the queries in the format '`?`' X Y — ask the distance (in definitions of the task) from the labeled point with coordinates (X, Y) to the point A ($-10^6 \le X, Y \le 10^6$, X and Y are integer).

If you are ready to print the answer, use the following format: '`!`' X Y (X and Y are the coordinates of the point A, then exit the program. This action is not considered as a query.

Examples

standard input	standard output
1 0	? 0 0 ? 1 0 ! 1 0

Note

For the correct interaction print the end-of-line after each query and after the answer and flush the output buffer with the respective functions of your programming language:

- `cout.flush()` or `fflush(stdout)` for C/C++;
- `stdout.flush()` for Python;
- look your language documentation for the other languages.

Problem E. Expression

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given positive integer N in its decimal representation without leading zeroes. Your task is to place **between** some digits the operator signs '+' (addition), '-' (subtraction) and '*' (multiplication, such that the following properties are held:

- Between any two digits can be placed no more than one operator, for example, the integer 1234 can be transformed into 12+34=56 or 1+2*3*4=25, but cannot be transformed into 1*-2+3+4 .
- You are not allowed to use the operator at the position zero, i.e. unary plus and minus are not allowed.
- The priority is defined as usual (i.e. the multiplication is peformed first). You are not allowed to use the brackets.

Your task is to place operators (or do not place them at all) to obtain the **least** possible result.

Input

The first line of the input contains one integer N ($1 \leq N < 10^{100}$).

Output

Print one integer — the minimal possible integer that can be obtained from N in a way descrobed in the problem statement.

Examples

standard input	standard output
9	9

Problem F. Find The Lap Length

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

You are controlling the robot that is placed on the stadium with an integer length. Initially robot is placed at the starting point.

The robot accepts the command "`run` k", that orders it to move along the distance exactly k meters counterclockwise from its current location and report total number of whole laps it passed while performing this and all previous commands (i.e. number of times the robot returned to the starting point after the start of race).

Your task is to determine the lap length using no more than 100 commands.

Interaction Protocol

The interaction is started by your program sending the line containing the query "`run` k" ($1 \le k \le 10^9$) where k is the distance that robot shall pass. Then the jury program prints one integer — total number of the whole laps that are passed after robot ends the movement (the lap is considered passed if the robot is moved through the starting point or stops in it after end of command).

You may assume that the lap length is integer between 1 and 10^9, inclusively.

You may use no more than 100 "`run`" commands.

If you are ready to print the answer, use the command "`ensure` s" where s is the lap length. After printing the answer the program shall terminate immediately.

For the correct interaction print the end-of-line after each query and the "`ensure`" command and flush the output buffer with the respective functions of your programming language:

- `cout.flush()` or `fflush(stdout)` for C/C++;
- `stdout.flush()` for Python;
- look your language documentation for the other languages.

Example

standard input	standard output
1 1 2 3	run 5 run 2 run 4 run 1 ensure 4

Problem G. Good Order

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

A network at the office consists of several nodes and bidirectional cables, connecting them with the next properties.

- Each cable is connecting exactly two nodes.
- Cable cannot connect the node with itself.
- Any two nodes are connected directly with no more than one cable.
- The information between any two nodes can be passed through one or more cables and intermediate nodes (i.e. any two nodes are linked by this network).

CEO wants the network to be in the **good order**, namely, he wants to have exactly one node with exactly one cable connected to it, exactly two nodes with exactly two cables connected to it and so on till N, i.e. for any $1 \le i \le N$ the network must contain exactly i nodes with exactly i cables connected to it. Sergeant does not allow any other nodes and cables except for that.

Your task is to build good order network for given integer N or consider that it is impossible to do it.

Input

Input contains one integer N ($1 \le N \le 239$) — the parameter of the network.

Output

If it is impossible to build good order network for given N, print -1. Otherwise, list all the cables in the network. Each cable must be listed at the new line, description of one cable must contain two 1-based indices of the nodes connected by the cable.

Example

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

Problem H. Highways and King

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are n towns conveniently labeled with $1, 2, \ldots, n$ in Flatland Kingdom and m bidirectional highways are allowed to be built.

The i-th highway can be built between cities a_i and b_i with cost c_i.

The Flatland Road Company plans to build the $(n-1)$ allowed highways with the least total cost to connect any of two cities directly or indirectly.

The King of Flatland is going to remove some of the highways from the list of the allowed highways. He would like to know the minimum number of highways to be removed to **strictly increase** the total cost.

Note that the total cost is considered as $+\infty$ if no valid $(n-1)$ highways exist after removing. It is also counted as ¡¡total cost strictly increases¿¿.

Input

The input contains zero or more test cases and is terminated by end-of-file. For each test case:

The first line contains two integers n and m. The i-th of the following m lines contains a_i, b_i, c_i.

- $2 \le n \le 50$
- $n - 1 \le m \le n^2$
- $1 \le a_i, b_i \le n$
- $1 \le c_i \le 10^9$
- Any two cities will be connected if all m allowed highways are built.
- The sum of n does not exceed 10^3.

Output

For each case, output an integer which denotes the result.

Example

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

Problem I. Jackpots

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

The rules of the final round of King's Lottery in Byteland are simple:

- The player starts to draw balls from an opaque bag containing a red balls, b green balls and c blue balls initially.
- After the player draws the ball, the ball is **not** returned into the bag.
- The player continues the process until he collects all a red balls, or all b green balls, or all c blue balls.
- The player wins the Gold Jackpot if he collects a red balls, Silver Jackpot if he collects b green balls, or Bronze Jackpot otherwise.

Given a, b and c, calculate the probabilities of winning each jackpot.

Input

The input contains three integers a, b, and c ($1 \le a, b, c \le 10^3$).

Output

Print three irreducible fractions p_G, p_S and p_B denoting the probability of winning the Gold Jackpot, the Silver Jackpot and the Bronze Jackpot, respectively. The fractions shall be printed in the form p/q.

Examples

standard input	standard output
1 1 1	1/3 1/3 1/3
1 2 3	7/12 4/15 3/20

Problem J. K-variance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider the *variance* of the sequence $a_1, a_2, \ldots, a_n$ as

$$\sqrt{\frac{\sum_{i=1}^{n}(a_i - \bar{a})^2}{n-1}}$$

where

$$\bar{a} = \frac{\sum_{i=1}^{n} a_i}{n}.$$

Consider the *K-variance* as the variance of the consecutive subsequence of length k.

Your task is to calculate all $(n - k + 1)$ K-variances for the given sequence and k.

Formally, the i-th $(1 \le i \le n - k + 1)$ K-variance r_i is the variance of sequence $\{a_i, a_{i+1}, \ldots, a_{i+k-1}\}$.

Input

The first line of the input contains 2 integers n, m $(2 \le m \le n \le 10^5)$.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ $(|a_i| \le 100)$.

Output

Print $(n - k + 1)$ lines with floating numbers $r_1, r_2, \ldots, r_{n-k+1}$.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}.

Examples

standard input	standard output
3 2 1 3 2	1.41421356 0.70710678
5 3 1 3 2 4 5	1.00000000 1.00000000 1.52752523

Problem K. Longest Bicolored Sequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given the stripe $1 \times n$ cells. Each cell is colored by one of 10^4 colors. The cells are enumerated from leftmost to rightmost one by sequential integers between 1 and n, inclusively.

Your task is to find the longest **continuous** sequence of the cells such that total number of colors in that sequence does not exceed 2. If there are more than one longest sequence, print the leftmost of them.

Input

The first line of the input contains one integer n — the length of the stripe ($1 \le n \le 10^5$). The second line contains n integers c_i ($1 \le c_i \le 10^4$) — the colors of the cells, listed from leftmost one to rightmost one. The cells of the same color are denoted by the same integer, the cells that have distinct colors — by different integers.

Output

Print two integers — the number of the cell where the longest continuous sequence that contains the cells of no more than two colors starts, and the length of this sequence. If there are more than one solutions, print one with the smallest first value (i.e. the leftmost one)

Examples

standard input	standard output
5 2 2 2 2 2	1 5
5 10 8 3 600 6	1 2
8 3 3 3 2 1 1 1 1	4 5

Problem L. Maximize It!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given two integers n and m, find the real α, such that $f(\frac{1}{2} + \alpha)$ is maximized, where $f(t) = \min_{i,j \in \mathbb{Z}} |\frac{i}{n} - \frac{j}{m} + t|$.

It can be proven that the value is rational. Print it as the irreducible fraction.

Input

The input data contains no more than 10^4 test cases, following one after another and terminated of the end-of-file,

Each test case consists of one string, containing two integers n and m ($1 \le n, m \le 10^9$).

Output

For each test case, print an irreducible fraction p/q ($q > 0$) — the maximum value of $f(\frac{1}{2} + \alpha)$.

Examples

standard input	standard output
1 1 1 2	1/2 1/4

Note

In the first test case $\alpha = 0$ maximizes the function.