Problem A. Game

Input: standard input
Output: standard output
Memory limit: 16 Mb

Watson and Rybka decided to play the following game. Players have ***N*** registers, each can store non-negative integer. At each move player should select one of the registers and subtract 2 or 3 from its value. Number in the register should remain non-negative. The player, who cannot make its move, loses the game. Players make moves alternately.

Watson and Rybka are very smart programs and always make moves optimally. You need to determine a winner. Rybka performs first move.

Input

The first line contains one integer N, following N lines contain A_i - initial values in registers.

$0 <= N <= 10^6$

$0 <= A_i < 10^9$

Output

You need to print winner's name – «*Watson*» or «*Rybka*» (without quotes).

Sample input	Sample output
3 2 3 4	Rybka
4 2 3 4 9	Watson

Problem B. Watson's memory

Input: standard input
Output: standard output
Memory limit: 16 Mb

Watson's memory is in critical state, it means all its cells are filled with ones. Rybka learned that if one considers all the Watson's memory as one big hexadecimal number, this number would be divisible by 7. However, she did not believe this and decided to check it. In order to do this, Rybka asked Watson about a number of cells his memory consists of. It turned out that the memory contains many cells – Rybka has not seen such a big numbers before. Help Rybka to check divisibility property of Watson's memory content.

Input

One non-negative decimal integer – number of cells in Watson's memory. This integer has not more than 100 decimal digits.

Output

You need to print checking result as one word (without quotes): «yes» - Watson's memory is divisible by 7, «no» - Watson's memory is not divisible by 7.

Sample input	Sample output
9	yes

Hint

Watson's memory is hexadecimal 111111111. This equals to decimal 4581298449, which is divisible by 7.

Problem C. Accomodation

Input: standard input
Output: standard output
Memory limit: 64 Mb

Watson is responsible for distribution of patients in a ward of a hospital. The ward contains ***N*** places, which are lined up in a row and sequentially numbered from 1 to ***N***. To minimize morbidity patients should be placed as far from each other or ward's entry as possible, and it is not allowed to relocate them. Watson has access to admission and discharging log. For each admission, Watson needs to find a place for new patient's allocation. In order to do this, Watson is to select the most distant from other patients place. If there are several such places, he should minimize the number of patients located closest to the new patient. If there are still several places, he should choose the most distant from entry place.

Input

The first line contains 2 integers: N – numbers on places in the ward and *Q* – number of entries in the log. Following *Q* lines contain entries from the log: «0» – admission, «1 X» - discharging of the patient which has been admitted according to the log's entry number X (numeration is 1-based and includes both admissions and dischargings).

It is guaranteed that the patient that is being discharged is present in the hospital. It is also guaranteed that for every admission there will be at least one place available.

$0 <= N, Q < 10^5$.

Output

You need to print place number for every admission in the log.

Sample input	Sample output
6 6 0 0 0 0 1 1 0	6 1 4 3 6

Problem D. Work

Input: standard input
Output: standard output
Memory limit: 16 Mb

Watson is doing important work. He randomly chooses subsequence from the sequence 1, 2, … ***N*** of length ***M***. Rybka wants to join the work, but Watson refused to share selected numbers, he just agreed to say for each number whether it is prime or not. To start a work, Rybka needs to find out all the numbers from the subsequence. Your task is to calculate how many numbers from the subsequence Rybka is able to determine uniquely, based on the information provided.

Input

The first line contains two integers ***N*** and ***M***. The second line contains ***M*** letters – description of the subsequence. Letter Y at ***i***-th place means that ***i***-th number is prime, N – the number is not prime. ***0 <= N, M < 3 * 10^4 (M <= N).***

Output

You need to print one number – count of subsequence elements that Rybka can determine.

Sample input	Sample output
5 3 YYY	3
7 3 NNN	3
6 3 YNY	2
8 3 NNN	0

Hint

In the first sample, there are exactly 3 primes: 2, 3, 5

In the second sample, there are 3 integers which are not prime: 1, 4, 6

In the third sample, the first number can be either 2 or 3, the second number is 4, and the third should be 5

In the fourth sample, following options are available: 1, 4, 6; 4, 6, 8, etc. We cannot determine number at any position in the subsequence uniquely.

Problem E. Construction

Input: standard input
Output: standard output
Memory limit: 16 Mb

Watson needs to construct most effective neural network. He is provided with a board containing ***N*** chips; chips are located in a line at distance 1 cm from each other. Each chip has its own performance, expressed by an integer. Watson is able to construct any number of neurons using these chips; every chip can be used not more than in one neuron. Neuron can be obtained by connecting two chips, which are located at distance ***M*** cm one from another. Effectiveness of a neuron is a sum of performances of chips it consists of. Effectiveness of neural network is a sum of effectivenesses of neurons. Neural network can contain any number of neurons, even 0.

Input

The first line contains two integers ***N*** and ***M***.

The second line contains ***N*** integers A_i – performances of chips.

$1 <= N, M <= 10^5$, $-10^6 < A_i < 10^6$

Output

The highest effectiveness of neural network built on the board with chips.

Sample input	Sample output
8 3 3 1 4 1 5 9 2 6	28

Hint

We can link chips in pairs in this way: 1-4, 3-6, 5-8.

Which will give effectiveness: (3 + 1) + (4 + 9) + (5 + 6).

Problem F. Task

Input: standard input
Output: standard output
Memory limit: 16 Mb

Watson set a simple task for Rybka – to find a sum of integers less than ***N***, which are divisible by ***A*** or ***B***, and print this sum modulo 1000000007 (***10^9 + 7***). Help Rybka to solve this task.

Input

One line contains three integers ***N, A,*** and ***B***.

1 <= N, A, B < 10^{18}.

Output

Print requested sum modulo 1000000007.

Sample input	Sample output
7 2 3	15

Hint

Numbers, divisible by 2: 2, 4, 6. Numbers, divisible by 3: 3, 6.

Sum of these numbers: 2 + 3 + 4 + 6 = 15

Problem G. Palindrome

Input: standard input
Output: standard output
Memory limit: 16 Mb

Rybka unsuccessfully tried to explain Watson what palindrome is, but he understands only numbers. Then Rybka decided to give Watson palindrome number. Number is palindrome if it can be read in the same way left to right and right to left. However, to make Watson's life funnier, she wants to give this number as 2 multipliers. You need to find the biggest palindrome number, which is a product of two numbers, each less than ***N***.

Input

One integer ***N.***

1 <= N <= 1000.

Output

You need to print two integers, multipliers of the biggest palindrome number (first smaller one, then bigger). If there several possible pairs, print the pair with the smallest first number.

Sample input	Sample output
100	91 99

Problem H. Control chain

Input: standard input
Output: standard output
Memory limit: 64 Mb

Rybka decided to measure criticality of Watson's control chain, which consists of ***N*** nodes, joined by conductors. Signal can be transmitted from any node to any other node trough the only possible path. When one of nodes comes under external action, it transmits signal to every other node. That is after external action the node creates (***N–1***) signals. ***i***-node intensity (*iNode*) of conductor is defined as product of its length to the number of signals, that go through it after external action to the node ***i***. The conductor is considered to be ***i***-critical, if its *iNode* is maximal among all other conductors (with fixed ***i***). If several equal *iNode* exist, several conductors are considered to be ***i***-critical. Total criticality of the conductor is defined as count of its ***i***-criticalities. Criticality of complete chain is defined as maximum of total criticalities of its conductors.

Input

The first line contains one integer ***N*** – number of nodes, following ***(N-1)*** lines describe conductors, each line contains 3 integers ***i, j, A*** – describes a conductor that connects nodes ***i*** and ***j*** and has length ***A***.

2 <= N <= 10^5, (1 <= i, j <= N, 1 <= A <= 10^5).

Output

You need to print criticality of the chain.

Sample input	Sample output
5 1 2 3 2 3 1 3 4 1 3 5 1	3

Hint

In the chain: 1-critical conductor 1; 2-critical: 1, 2; 3-critical: 1; 4-critical: 3; 5-critical: 4. Hence, the first conductor has 3 criticalities, so its total criticality is 3, total criticality of other conductors is 1, criticality of the chain equals to maximum of conductors' criticalities, that is 3.

Problem I. Cube

Input: standard input
Output: standard output
Memory limit: 128 Mb

Watson has a cube, which is placed on an infinite plane. We will use Cartesian coordinate system on it. Center of cube's bottom face is located at point *(0, 0)*, all its edges are either parallel or orthogonal to the coordinate axes. We will number its faces from 0 to 5, in the way that sum of numbers at opposite faces equals 5, and if we look at the cube along ***Oy*** axe, we will see face 0, while top and right faces will have numbers 1 and 2 correspondingly. We say that this orientation is *(0, 1, 2)*-orientation of the cube on the plain. Watson is to move the cube to coordinates ***(x, y)*** and put it into ***(a, b, c)***-orientation. Cube can be rolled left, right, forward or backward, giving control commands **L, R, U, D**:

L – roll left (x coordinate is decreased by 1),

R – roll right (x coordinate is increased by 1),

U – roll forward (y coordinate is increased by 1),

D – roll backward (y coordinate is decreased by 1).

Watson is going to do as little operations as possible, so he wants to build a sequence of control commands as short as possible. If several sequences are possible, he chooses lexicographically smallest one. Certainly, Rybka is assigned to move the cube. Watson started to transfer commands to Rybka, and she started executing it. When ***N*** commands have been executed, Watson was drawn away by important surgery. Still Rybka does not want to stop and plans to execute additional ***K*** commands. Help Rybka to find out necessary sequence of commands.

Input

The first line contains coordinates ***x*** and ***y***, not exceeding 10^9 by absolute value.

The second line contains numbers describing required ***(a, b, c)***-orientation ($0 <= a, b, c < 6$).

The third line contains integers ***N*** and ***K*** – number of executed and planned commands ($0 <= N <= 10^9$, $1 <= K <= 1000$).

Output

You need to print «***-1***» (without quotes), if final state is not reachable, otherwise print sequence of ***K*** characters {**L, R, U, D** or –}.

Character «-» means that it is not necessary to move (final state has been reached).

Sample input	**Sample output**
1 1 0 1 2 0 10	DRULUR----
30 -40 0 1 0 10000 23	-1

Problem J. Duty

Input: standard input
Output: standard output
Memory limit: 16 Mb

Ministry of health decided to sanitize streets of city D. In order to do it a crew of highly skilled doctors has been formed. City has ***N*** hospitals, which are connected by ***M*** paths. For sanitization of paths between hospitals, the crew needs to go through every path exactly once and return to the original point. Since it's not always possible, crew can be teleported. Teleportation is quite expensive for the ministry operation, so it should be used as seldom as possible. There can be many paths between same hospitals, same as there could be paths that connect hospital to itself. Crew can start its path at any hospital. Difficult task to minimize expenses has been assigned to Watson by Ministry. He has activated all its neural chips and dove into calculations. Rybka volunteered to help and find simpler way to solve the task.

Input

The first line contains two integers ***N*** and ***M*** (***1 <= N, M < 10^5***).

Following ***M*** lines each contains two integers ***i j***, describing path between hospitals ***i*** and ***j*** (***1 <= i, j <= N***).

Output

You need to print minimal possible number of teleportations.

Sample input	Sample output
4 6 1 2 2 3 3 4 4 1 1 3 2 4	2

Problem K. Method of linear transformation

Input: standard input
Output: standard output
Memory limit: 16 Mb

Watson is assigned to perform diagnosis of cardiovascular diseases, and he needs to process ECGs of patients. He applies linear transformation method to the signal. Shape of the signal can be described by formula:

$$F(x) = \int_{-\infty}^{\infty} K(x - x')\psi(x')dx',$$

where $\psi(x')$ - function, that describes position and intensity of particular components of the signal , $K(x) = \frac{A}{\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$ - a Gaussian function describing the shape and width of particular spectral lines. Coefficients $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ Watson gets as an input, coefficient $\boldsymbol{A}$ here and further is calculated by formula $A = \frac{1}{\sqrt{2\pi}}$.

Definition of function $\psi(x)$ can be given in this way: $\psi(x) = A\int_{-\infty}^{\infty} \frac{\tilde{F}(y)}{\tilde{K}(y)} \exp(ixy)dy$,

where Fourier transform of function $\boldsymbol{F(x)}$ is calculated as: $\tilde{F}(y) = A\int_{-\infty}^{\infty} F(x)\exp(-iyx)dx$

To make exact computations in diagnosis tasks Watson needs to know numeric value for the coefficient $\boldsymbol{A}$.

Input

The only line at input contains two positive numbers $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, each not exceeding 1000.

Output

You need to print coefficient $\boldsymbol{A}$ with 4 digits after decimal point.

Sample input	Sample output
1 2	0.3989

Problem L. Watson's magic number

Input: standard input
Output: standard output
Memory limit: 16 Mb

Rybka wants to know Watson's magic number. Watson's magic number is a determinant of Watson's magic matrix. Watson's magic matrix is some known matrix with dimensions 3x3.

Input

Input contains three lines, each line contains three integers less than 100, which defines Watson's magic matrix.

Output

You need to print Watson's magic number.

Sample input	Sample output
1 2 3 2 3 4 2 1 1	-1

Problem M. Formatting

Input: standard input
Output: standard output
Memory limit: 16 Mb

Rybka is going to format Watson's hard drive partition. This partition should contain ***N*** sectors, sequentially numbered with decimal integers starting from ***A***. How many digits Rybka needs to enumerate all sectors?

Input

Positive integers ***N*** and $A < 10^8$

Output

Number of digits necessary for formatting

Sample input	Sample output
3 9	5

Hint

Three numbers are used: 9, 10, 11. These numbers contain 5 digits.