

Problem A. A lazy controller

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

One of the shops of N-sky machine-building plant produces spare parts for cars. In order to monitor the quality, the weight of every detail is to be controlled. For that there is a special controlling person in charge. Unfortunately, the controller in this checking unit is rather lazy, and instead of weighing every detail separately, he decides to «optimize» the process a little. Also all similar (of the same type) parts are supposed to be of the same weight.

At the end of the working day, a box with many identical parts arrives for inspection. The controller knows that a quality part should weigh a integer number of grams, but is too lazy to look into the documents and find out how much exactly.

The «optimized» control process is as follows:

1. The controller carries out k weighings, and numbers them from 1 to k .
2. For the i -th weighing, the number a_i is randomly chosen for the number of parts; this number is recorded in the weights register-book.
3. The selected number of parts is placed on the scales, and their total weight w_i is also recorded in the register-book.
4. Upon completion of k weighing, the register-book is analyzed and a conclusion is made on the presence or absence of defects.

Naturally, such an «optimized» process does not always reveal the presence of faulty spare parts, and does not indicate a specific defective part. The controller is so lazy that in the event of signs of a defective item rejects the entire batch. In all other cases, the batch receives a sign of quality.

Your task will be to write a program that, according to the weights register-book, will determine whether the batch will receive a quality mark or will be rejected.

Input

The first line of the input file contains an integer k ($1 \leq k \leq 1000$). Next follow k number of lines, each of which contains integers a_i ($0 < a_i \leq 1000$) and w_i ($0 < w_i < 1000000$), separated by a space.

Output

The output line should contain the word «DEFECT» if there are for sure defective parts in the batch, or «QUALITY» if there are no signs of defective parts.

Examples

standard input	standard output
3 10 30 5 15 2 6	QUALITY
2 2 5 4 10	DEFECT

Note

The participant must display «QUALITY» in the event that there are no definite signs of defect, even if there is not enough data for an unambiguous conclusion about the quality of the details

Problem B. Gremlins attack!

Input file: `standard input`
Output file: `standard output`
Time limit: 4 seconds
Memory limit: 64 megabytes

There's an emergency in a county-level American town. Gremlins have scattered around the town and thus the rest of the world is in danger, as, at least, one of them seems to have left the town. Special intelligence services are currently restoring the course of events and trying to identify the earliest moment when it could happen.

The town is represented in the form of a square checkered board of size $N \times N$. Each cell represents a single house. Gremlins are, typically, afraid of bright light and therefore tend to hide in darkness. Furthermore, as people go to bed, the lights in the houses are going out. Gremlins can move horizontally and vertically from one house, where the light is off, to another. As soon as they reach the house which is on the border of the town, they escape from the town.

You have reliable data about which houses gremlins occupied in the beginning, as well as the history of turning the lights off in the houses. You need to display number of the house in history, from which after turning the light off, at least, one gremlin could escape from the town. If gremlins can escape from the town immediately, type 0. The numbering of houses in history starts with one.

Input

In the first line, three integers are entered, separated by spaces, $1 < N \leq 500$, $1 \leq M \leq N^2$, $1 \leq K \leq N^2$, where N determines the size of the city, M is the number of houses in which gremlins are located at the beginning, K is the size of the history of turning off lights in the houses.

Then there are M lines, each of which contains a pair of numbers $0 \leq x_i, y_i < N$, specifying the coordinates of the houses where gremlins are located at the beginning. After that, there follow K lines, each of which contains a pair of numbers $0 \leq x_j, y_j < N$, specifying the coordinates of the houses in which the light is turned off.

Output

The single number from 0 to K , which sets the number of the house, in which after turning off the light, at least, one gremlin had the opportunity to escape from the town.

Examples

standard input	standard output
3 1 3 1 1 0 0 0 1 0 2	2
5 2 5 0 1 4 1 0 0 1 1 2 2 3 3 4 4	0
4 2 3 1 1 1 2 2 0 3 1 1 3	3
5 2 6 1 1 3 3 1 2 1 3 2 3 3 0 3 1 2 1	6
7 6 7 1 4 1 1 2 3 3 1 4 4 5 2 0 4 2 4 3 4 1 0 2 1 5 1 5 0	1

Note

The input data guarantee that the escape could have been made.

Problem C. Dimensions

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

When performing various physical calculations, it is often necessary to deal not only with numbers, but also with the dimensions of quantities. In this case, the dimensions can be very complex and they need to be reduced: for example, the dimension $\text{kg} / (\text{kg} / \text{m} / \text{s})$ is just m / s .

Write a program that minimizes the dimension as much as possible and registers all occurring values from left to right in alphabetical order (if the dimension is a fraction, then the numerator and denominator are written in alphabetical order separately).

Input

The first line of the input file contains a single character string - the dimension to be reduced.

Dimensions are denoted by Latin lowercase and uppercase letters.

In addition to the Latin letters in the expression may be signs of multiplication «*», division «/» and brackets. The degrees of magnitudes are expressed as multiple repetitions of magnitude. The order of operations corresponds to the generally accepted. Values differing by letter case are considered different.

The length input string does not exceed 1000 characters.

Output

The output contains a reduced dimension. The first line contains the numerator of the dimension, the second - the denominator. It is possible to use the 1 in the form of the numerator and / or denominator.

Examples

standard input	standard output
kg/(kg/(m/s))	m s
kg*a/(Fs*B)*A*Kt	A*a*Kt*kg B*Fs

Note

Sorting in alphabetical order implies the following order of letters: «AaBbCbDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz». Shorter lines come earlier, than longer lines.

Problem D. We were trying to share an orange ...

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

As you know, oranges are made up of slices, and it is quite convenient to divide them into several portions for several people. Obviously, anyone can eat it up by oneself, or divide it for a company in which there are as many people as slices in the orange itself.

If the orange slices can be equally divided among n people, then we say that it is suitable for such a company. And what minimum number of slices m should be in an orange, so that it could be suitable for k different companies (companies are considered different if they consist of a different number of people)?

Write a program which, according to the number of companies k , will find the minimum number of orange slices suitable for exactly these different companies k or report that such a number does not exist.

Input

The input file contains a single integer k ($1 \leq k \leq 1000$).

Output

The output file contains a single integer m or -1 if such a number does not exist.

Examples

standard input	standard output
4	6
1	1

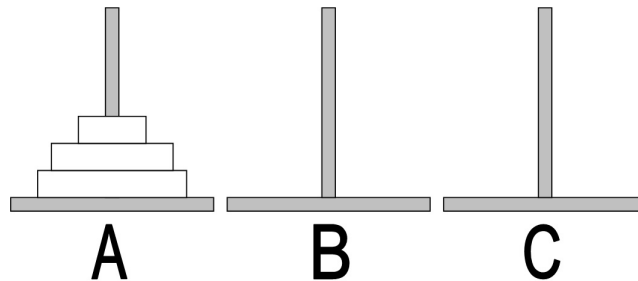
Note

An orange consisting of 6 slices can be equally divided between 1, 2, 3 and 6 persons.

Problem E. Hanoi Tower

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

It has become a good tradition in programming competitions to solve the problem of Hanoi Tower.



Let us briefly recall the rules of this puzzle game.

There are three rods: *A*, *B*, *C*. Initially, *N* disks of different diameters are placed on the source rod *A*: the smallest disk is at the top, and then come other disks as the diameter increases. The second and third rods are empty so far.

You want to move all the disks from the source rod *A* to the resultant rod *B*, using the third rod *C* as an auxiliary. In one move, it is allowed to take one (necessarily upper) disk and put it on another rod, possibly empty. Moreover, a disc of a larger diameter can not be placed on a smaller disc.

Many programming textbooks provide a solution to this puzzle.

```
Procedure Hanoi (X, Y, Z: char; N: integer);
Begin
  If N>0 then
    Begin
      Hanoi (X, Z, Y, N-1);
      Writeln('Disk ', N, ' from ', X, ' to ', Y);
      Hanoi (Z, Y, X, N-1)
    End
  End;
End;
```

Associate Professor P. from the city of R., preparing for the programming championship, decided to warm up and began to manually shift the disks of Hanoi Tower. (For this, he used rings from a children's pyramid and three rods taken from his granddaughter.)

At some point, the granddaughter, offended by her grandfather, selected a part of the rings, without touching the source rod with the strung discs-rings.

Help the assistant professor and calculate the minimum possible number of moves he has managed to make if the current location of the disks on the source rod is known.

Disk movements are performed using the standard algorithm described in the procedure above. The combination of disks specified in the input file is guaranteed to be achievable.

Input

The first line contains a single integer *N* - the initial number of disks ($1 \leq N \leq 100$).

The second line of the file contains *N* characters '0' and '1'. The symbol in the *i*-th position indicates whether a disk with the number *i* is present (specified by the '1' sign) or not (specified by the '0' sign) on the initial rod at the moment of interruption into the game.

Output

The output file must contain one integer in binary form without the preceding zeros - the minimum possible number of moves, after which the disks on the source rod are placed in a specified way.

Examples

standard input	standard output
3 111	0
3 001	10
5 00000	10000

Problem F. Pebbles

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

At the computer science exam, Vasya was solving the following problem:

«Two players, Petya and Vanya, are playing the following game. In front of the players there is a pile of stones. Players take turns, Petya is the first to play. In one move, the player can add a stones to the pile or increase the number of stones in the pile twice. For example, if $a = 2$, having a pile of 10 stones, in one move you can get a pile of 12 or 20 stones. Each player has an unlimited amount of stones to make moves.

The game ends at the moment when the number of stones in the pile becomes at least n . The loser is the player who made the last move, that is, who received a pile of n or more stones.

At the initial moment there were s stones in the pile. It is said that a player has a winning strategy if he can win in any opponent's moves.»

Unfortunately, Vasya does not remember the values of a and n , which were at the exam, but he remembers at which initial s Petya or Vanya won. He also remembers that a , n and s did not exceed 10^3 .

Write a program which, using different values of s and a winning strategy for these s , will find the minimum suitable a and n .

Input

The first line of the input file contains the number k — the number of positions ($1 \leq k \leq 1000$) for which Vasya remembers the outcomes.

Then follow k lines, each of which contains two integers. The first of these numbers is the initial number of stones s_i ($0 < s_i \leq 1000$), the second is 1, if with the number of stones s_i , Petya wins, or 2 if with this number of stones Vanya wins. All initial positions of s_i are different.

Output

The output file must contain two integers n and a , separated by a space ($0 < n, a \leq 1000$).

If the problem allows several solutions, then a solution with a minimum n must be derived; if there are several solutions for a minimum n , then a solution with minimal n and a must be derived.

If the solution does not exist, then two zeros should be output.

Examples

standard input	standard output
4 1 2 2 1 3 1 4 2	5 1
2 1 2 2 2	0 0

Problem G. Non-random numbers

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

For generating pseudo-random numbers, various methods are used. One of such methods is the use of Boolean recurrent expressions. In such a case, the pseudo-random number is represented as a bit sequence, such that each next bit is calculated from the previous ones using a certain formula. However, such a task actually determines the sequence of its first members.

An alternative is the use of Boolean equations. We will say that the bit sequence $x_1, x_2, x_3, \dots, x_n$ satisfies some equation

$$F(x_i, x_{i-1}, \dots, x_{i-k}) = 1,$$

if this equation is true for any $x_i, x_{i-1}, \dots, x_{i-k}$, for $i > k$. Write a program that, using a given equation, will find the number of sequences of length n that satisfy this equation.

Input

The first line of the input file contains an integer n ($2 \leq n \leq 10^3$).

The second line contains the left side of the equation, encoded in the following way:

- the bit with the index x_{i-r} ($0 \leq r \leq 9, r < n$) is denoted as r , that is, x_{i-5} will be denoted as 5.
- operation «and» (conjunction) is denoted as «&»
- the operation «or» (disjunction) is denoted as «|»
- the operation «exclusive or» (addition on module 2) is denoted as «+»
- the operation «not» (negation) is denoted as «-»
- brackets are allowed in the equation, operations in brackets are executed earlier than others
- binary operations that are not in brackets are considered peer-to-peer and are performed strictly from left to right
- spaces inside the relationship are not allowed

The length of left side of the equation is not longer 1000 characters.

Output

The output file contains a single number — the number of bit sequences of length n that satisfy the equation. The number should be modulo 2^{60} .

Examples

standard input	standard output
3 (0+2) (0&2)	6
4 (0+2) - (0&2)	9

Note

The first example encodes the equation $(x_i + x_{i-2})|(x_i \& x_{i-2}) = 1$, which satisfies the following sequence: 001, 011, 100, 101, 110, 111.

The second example encodes the equation $(x_i + x_{i-2})|(-(x_i \& x_{i-2})) = 1$, which is satisfied by the sequence: 0000, 0001, 0010, 0011, 0100, 0110, 1000, 1001, 1100.

Problem H. A self-describing sequence

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Let's call the sequence $a_0, a_1, a_2, \dots, a_{k-1}$ as «self-describing»; if the value a_i is the number of numbers i encountered in this sequence.

For example, in a sequence 1, 2, 1, 0 - «0» occurs 1 time, «1» is found 2 times, «2» occurs once and «3» does not occur.

Your task will be to write a program that for a given length of sequence k , will output its elements with the specified indices, or report that such a sequence does not exist. If there are several such sequences, any of them is considered to be correct.

Input

The first line of the input file contains the length of the sequence k . ($1 \leq k \leq 2^{30}$).

The second line of the input file contains the number n ($0 < n \leq 10^5$) which is the number of elements to be found in the sequence.

The third line contains the n number of integers $r_1 r_2 r_3 \dots r_n$ ($0 \leq r_i < k$), separated by spaces. These are the indices of those elements of the sequence which are to be put out.

Output

The first line of the output file contains 0 if the «self-describing» sequence of length k does not exist. Otherwise, the first line contains the number n , further, in the second line of the output file it is situated by the n numbers separated by spaces — elements of the sequence with the indices specified in the input file in the order of appearance of these indices in the input file.

Examples

standard input	standard output
4 4 0 1 2 3	4 1 2 1 0
4 4 3 2 1 0	4 0 1 2 1
1 1 0	0
5 3 4 4 4	3 0 0 0

Problem I. Noughts and crosses

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

There are many options for playing noughts and crosses. Let's consider one of them.

A number of crosses and noughts are placed on the checkered square board, and there are some empty cells left. The player must fill in the empty squares so that neither crosses nor noughts form lines of more than three identical signs in a row. Lines are counted horizontally, vertically, or of diagonally format.

Your task will be to write a program which, in a given initial field, fills all empty cells with crosses or noughts according to the rules described above. If there are several such arrangements, then any of them will be considered correct.

It is guaranteed that for the input data, there is at least one correct solution.

Input

The first line of the input file contains number n — the size of the board ($3 < n \leq 20$). This is followed by n strings of n characters in each — the initial state of the playing field. The cross is indicated by a «+» (plus) sign, a nought is indicated by a «0» (zero) symbol, an empty field is indicated by a «.» (dot) symbol.

Output

The output format is identical to the input data format (including board size). The solution should not contain empty fields.

Example

standard input	standard output
4	4
0.0.	0+00
00+0	00+0
0+00	0+00
++..	++0+

Problem J. R u really ready?

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

You have two strings pattern p and text t . Strings consist of only latin lower case characters. In addition pattern may include characters $+$ which means that a previous symbol should be repeated one or more times and characters $*$ in that case previous symbol should be repeated any number times.

Your task is to answer on the question does text t matche to pattern p .

For example, text "*eboooy*" matches to patterns "*ebo+y*", "*ebo*o*o+y+*", but doesn't to "*ebb*oy*" or to "*ebooo*".

Input

The first string of input consists pattern p , the second one consists text t . Both strings have non-negative length. It's guaranteed that p is a valid in the meaning there are no two $+$ or $*$ in a row and the first symbol of p is latin lower case character. Each string can't be longer than 1000 characters.

Output

Print *Yes* if text matches pattern, otherwise print *No*.

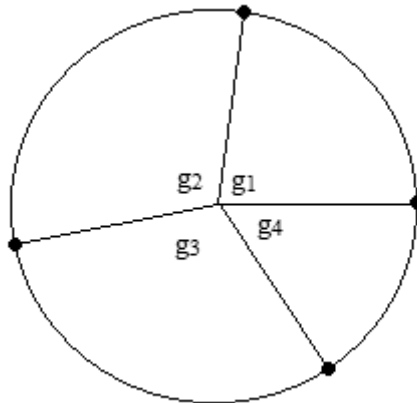
Examples

standard input	standard output
pa*t+ern pattern	Yes
c*cp+p cpp	Yes
b+b b	No

Problem K. Meson Collider

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Four spark chambers are installed along the large meson collider, which monitors the interactions of mesons moving in a circle at high velocities.



In order to receive reliable synchronized data from spark chambers and process it quickly, it is required to place 2 computers inside the circle, connect them via a cable to each other, as well as to the spark chambers. Each camera must be connected to at least one computer. The task is to minimize the total length of the cable for maximum synchronization.

Write a program that calculates the cable length using the known location of the spark chambers.

Input

In the first line, five integers are entered, separated by spaces, R is the radius of the circle along which the mesons move ($1 \leq R \leq 100$), g_1, g_2, g_3, g_4 are the angles at which spark chambers are visible from the center of the circle ($60^\circ \leq g_i \leq 360^\circ$).

Output

Print the total length of the required cable. Your answer will be accepted if absolute or relative error does not exceed 10^{-6} .

Formally, let your answer be a , and the jury's answer be b . Your answer is considered correct if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Example

standard input	standard output
10 60 120 60 120	34.64101615

Note

Illustration for test

