**Python sprint – Task1**

**Problems:**

1.A magic number is a number formed by concatenation of numbers 1, 14 and 144. We can use each of these numbers any number of times. Therefore 14144, 141414 and 1411 are magic numbers but 1444, 514 and 414 are not.

You're given a number. Determine if it is a magic number or not.

Input

The first line of input contains an integer *n*, (1 ≤ *n* ≤ 109). This number doesn't contain leading zeros.

Sample : 144411

Output

Print "YES" if *n* is a magic number or print "NO" if it's not.

Sample : YES

2. You have decided to watch the best moments of some movie. There are two buttons on your player:

1. Watch the current minute of the movie. By pressing this button, you watch the current minute of the movie and the player automatically proceeds to the next minute of the movie.
2. Skip exactly *x* minutes of the movie (*x* is some fixed positive integer). If the player is now at the *t*-th minute of the movie, then as a result of pressing this button, it proceeds to the minute (*t* + *x*).

Initially the movie is turned on in the player on the first minute, and you want to watch exactly *n* best moments of the movie, the *i*-th best moment starts at the *li*-th minute and ends at the *ri*-th minute (more formally, the *i*-th best moment consists of minutes: *li*, *li* + 1, ..., *ri*).

Determine, what is the minimum number of minutes of the movie you have to watch if you want to watch all the best moments?

Input

The first line contains two space-separated integers *n*, *x* (1 ≤ *n* ≤ 50, 1 ≤ *x* ≤ 105) — the number of the best moments of the movie and the value of *x* for the second button.

The following *n* lines contain the descriptions of the best moments of the movie, the *i*-th line of the description contains two integers separated by a space *li*, *ri* (1 ≤ *li* ≤ *ri* ≤ 105).

It is guaranteed that for all integers *i* from 2 to *n* the following condition holds: *ri*- 1 < *li*.

Sample : 2 3

5 6

10 12

Output

Output a single number — the answer to the problem

Sample : 6

3. Jeff's got *n* cards, each card contains either digit 0, or digit 5. Jeff can choose several cards and put them in a line so that he gets some number. What is the largest possible number divisible by 90 Jeff can make from the cards he's got with?

Jeff must make the number without leading zero. At that, we assume that number 0 doesn't contain any leading zeroes. Jeff doesn't have to use all the cards.

Input

The first line contains integer *n* (1 ≤ *n* ≤ 103). The next line contains *n* integers *a*1, *a*2, ..., *an* (*ai* = 0 or *ai* = 5). Number *ai* represents the digit that is written on the *i*-th card.

Sample : 4

5 0 5 0

Output

In a single line print the answer to the problem — the maximum number, divisible by 90. If you can't make any divisible by 90 number from the cards, print -1.

Sample : 0

4. Valera is a little boy. Yesterday he got a huge Math hometask at school, so Valera didn't have enough time to properly learn the English alphabet for his English lesson. Unfortunately, the English teacher decided to have a test on alphabet today. At the test Valera got a square piece of squared paper. The length of the side equals *n* squares (*n* is an odd number) and each unit square contains some small letter of the English alphabet.

Valera needs to know if the letters written on the square piece of paper form letter "X". Valera's teacher thinks that the letters on the piece of paper form an "X", if:

* on both diagonals of the square paper all letters are the same;
* all other squares of the paper (they are not on the diagonals) contain the same letter that is different from the letters on the diagonals.

Help Valera, write the program that completes the described task for him.

Input

The first line contains integer *n* (3 ≤ *n* < 300; *n* is odd). Each of the next *n* lines contains *n* small English letters — the description of Valera's paper.

Sample : 3

w s w

s w s

w s w

Output

Print string "YES", if the letters on the paper form letter "X". Otherwise, print string "NO". Print the strings without quotes.

Sample : YES

5. A company has *n* employees numbered from 1 to *n*. Each employee either has no immediate manager or exactly one immediate manager, who is another employee with a different number. An employee *A* is said to be the *superior* of another employee *B* if at least one of the following is true:

* Employee *A* is the immediate manager of employee *B*
* Employee *B* has an immediate manager employee *C* such that employee *A* is the superior of employee *C*.

The company will not have a managerial cycle. That is, there will not exist an employee who is the superior of his/her own immediate manager.

Today the company is going to arrange a party. This involves dividing all *n* employees into several groups: every employee must belong to exactly one group. Furthermore, within any single group, there must not be two employees *A* and *B* such that *A* is the superior of *B*.

What is the minimum number of groups that must be formed?

Input

The first line contains integer *n* (1 ≤ *n* ≤ 2000) — the number of employees.

The next *n* lines contain the integers *pi* (1 ≤ *pi* ≤ *n* or *pi* = -1). Every *pi* denotes the immediate manager for the *i*-th employee. If *pi* is -1, that means that the *i*-th employee does not have an immediate manager.

It is guaranteed, that no employee will be the immediate manager of him/herself (*pi* ≠ *i*). Also, there will be no managerial cycles.

Sample : 5

-1

1

2

1

-1

Output

Print a single integer denoting the minimum number of groups that will be formed in the party.

Sample : 3

6. Robot Doc is located in the hall, with *n* computers stand in a line, numbered from left to right from 1 to *n*. Each computer contains **exactly one** piece of information, each of which Doc wants to get eventually. The computers are equipped with a security system, so to crack the *i*-th of them, the robot needs to collect at least *ai* any pieces of information from the other computers. Doc can hack the computer only if he is right next to it.

The robot is assembled using modern technologies and can move along the line of computers in either of the two possible directions, but the change of direction requires a large amount of resources from Doc. Tell the minimum number of changes of direction, which the robot will have to make to collect all *n* parts of information if initially it is next to computer with number 1.

**It is guaranteed** that there exists at least one sequence of the robot's actions, which leads to the collection of all information. Initially Doc doesn't have any pieces of information.

Input

The first line contains number *n* (1 ≤ *n* ≤ 1000). The second line contains *n* non-negative integers *a*1, *a*2, ..., *an* (0 ≤ *ai* < *n*), separated by a space. It is guaranteed that there exists a way for robot to collect all pieces of the information.

Sample : 3

0 2 0

Output

Print a single number — the minimum number of changes in direction that the robot will have to make in order to collect all *n* parts of information.

Sample : 1

7. Sergey is testing a next-generation processor. Instead of bytes the processor works with memory cells consisting of *n* bits. These bits are numbered from 1 to *n*. An integer is stored in the cell in the following way: the least significant bit is stored in the first bit of the cell, the next significant bit is stored in the second bit, and so on; the most significant bit is stored in the *n*-th bit.

Now Sergey wants to test the following instruction: "add 1 to the value of the cell". As a result of the instruction, the integer that is written in the cell must be increased by one; if some of the most significant bits of the resulting number do not fit into the cell, they must be discarded.

Sergey wrote certain values ​​of the bits in the cell and is going to add one to its value. How many bits of the cell will change after the operation?

Input

The first line contains a single integer *n* (1 ≤ *n* ≤ 100) — the number of bits in the cell.

The second line contains a string consisting of *n* characters — the initial state of the cell. The first character denotes the state of the first bit of the cell. The second character denotes the second least significant bit and so on. The last character denotes the state of the most significant bit.

Sample : 4

1100

Output

Print a single integer — the number of bits in the cell which change their state after we add 1 to the cell.

Sample : 3

8. The Berland Armed Forces System consists of *n* ranks that are numbered using natural numbers from 1 to *n*, where 1 is the lowest rank and *n* is the highest rank.

One needs exactly *di* years to rise from rank *i* to rank *i* + 1. Reaching a certain rank *i* having not reached all the previous *i* - 1 ranks is impossible.

Vasya has just reached a new rank of *a*, but he dreams of holding the rank of *b*. Find for how many more years Vasya should serve in the army until he can finally realize his dream.

Input

The first input line contains an integer *n* (2 ≤ *n* ≤ 100). The second line contains *n* - 1 integers *di* (1 ≤ *di* ≤ 100). The third input line contains two integers *a* and *b* (1 ≤ *a* < *b* ≤ *n*). The numbers on the lines are space-separated.

Sample : 3

5 6

1 2

Output

Print the single number which is the number of years that Vasya needs to rise from rank *a* to rank *b*.

Sample : 11

9. Your friend has recently learned about coprime numbers. A pair of numbers {*a*, *b*} is called *coprime* if the maximum number that divides both *a* and *b* is equal to one.

Your friend often comes up with different statements. He has recently supposed that if the pair (*a*, *b*) is coprime and the pair (*b*, *c*) is coprime, then the pair (*a*, *c*) is coprime.

You want to find a counterexample for your friend's statement. Therefore, your task is to find three distinct numbers (*a*, *b*, *c*), for which the statement is false, and the numbers meet the condition *l* ≤ *a* < *b* < *c* ≤ *r*.

More specifically, you need to find three numbers (*a*, *b*, *c*), such that *l* ≤ *a* < *b* < *c* ≤ *r*, pairs (*a*, *b*) and (*b*, *c*) are coprime, and pair (*a*, *c*) is not coprime.

Input

The single line contains two positive space-separated integers *l*, *r* (1 ≤ *l* ≤ *r* ≤ 1018; *r* - *l* ≤ 50).

Sample: 2 4

Output

Print three positive space-separated integers *a*, *b*, *c* — three distinct numbers (*a*, *b*, *c*) that form the counterexample. If there are several solutions, you are allowed to print any of them. The numbers must be printed in ascending order.

If the counterexample does not exist, print the single number -1.

Sample : 2 3 4

**Problem assignment:**

It is one question per two of you. The steps you need follow to upload is:

1.Create a folder with your name

2.Add two files “Problem.txt” and “Solution.txt” [Case sensitive]

3.Add your name in the “Names.txt” in the format [Name of the folder]-[Language used]

Don’t change the format under any cause. I use javascript to loop through, so please stick to it. All in branch ne.

1. Aadit Jain , Ashly Rose
2. Akshay Saini, Aishwarya
3. Vaibhav Gupta, Akshay Iyer
4. Ravi Kumbhkar, Nirmal
5. Kanika Narang, Akshit Sharma
6. Paartipabhalaji, Govindarajan Sivaraj
7. Surentharan, Sailesh
8. Rajesh, Kumar Akshay
9. Baranitharan and myself

Good luck!

Have fun coding!