A. Pasha and Pixels

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pasha loves his phone and also putting his hair up... But the hair is now irrelevant.

Pasha has installed a new game to his phone. The goal of the game is following. There is a rectangular field consisting of n row with m pixels in each row. Initially, all the pixels are colored white. In one move, Pasha can choose any pixel and color it black. In particular, he can choose the pixel that is already black, then after the boy's move the pixel does not change, that is, it remains black. Pasha loses the game when a 2×2 square consisting of black pixels is formed.

Pasha has made a plan of k moves, according to which he will paint pixels. Each turn in his plan is represented as a pair of numbers i and j, denoting respectively the row and the column of the pixel to be colored on the current move.

Determine whether Pasha loses if he acts in accordance with his plan, and if he does, on what move the 2×2 square consisting of black pixels is formed.

Input

The first line of the input contains three integers n, m, k ($1 \le n, m \le 1000$, $1 \le k \le 10^5$) — the number of rows, the number of columns and the number of moves that Pasha is going to perform.

The next k lines contain Pasha's moves in the order he makes them. Each line contains two integers i and j ($1 \le i \le n$, $1 \le j \le m$), representing the row number and column number of the pixel that was painted during a move.

Output

If Pasha loses, print the number of the move when the 2×2 square consisting of black pixels is formed.

If Pasha doesn't lose, that is, no 2×2 square consisting of black pixels is formed during the given k moves, print 0.

Examples

input
2 2 4 1 1 1 2 2 1 2 2
output
4

input
2 3 6 2 3 2 2 1 3 2 2 1 2 1 1
output
5

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

B. Anton and currency you all know

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Berland, 2016. The exchange rate of *currency you all know* against the burle has increased so much that to simplify the calculations, its fractional part was neglected and the exchange rate is now assumed to be an integer.

Reliable sources have informed the financier Anton of some information about the exchange rate of *currency you all know* against the burle for tomorrow. Now Anton knows that tomorrow the exchange rate will be an even number, which can be obtained from the present rate by swapping exactly two distinct digits in it. Of all the possible values that meet these conditions, the exchange rate for tomorrow will be the maximum possible. It is guaranteed that today the exchange rate is an *odd* positive integer n. Help Anton to determine the exchange rate of *currency you all know* for tomorrow!

Input

The first line contains an odd positive integer n — the exchange rate of *currency you all know* for today. The length of number n's representation is within range from 2 to 10^5, inclusive. The representation of n doesn't contain any leading zeroes.

Output

If the information about tomorrow's exchange rate is inconsistent, that is, there is no integer that meets the condition, print -1.

Otherwise, print the exchange rate of *currency you all know* against the burle for tomorrow. This should be the maximum possible number of those that are even and that are obtained from today's exchange rate by swapping exactly two digits. Exchange rate representation should not contain leading zeroes.

Examples

input
527
output
572

input
4573
output
3574

input
1357997531
output
-1

C. Anya and Ghosts

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Anya loves to watch horror movies. In the best traditions of horror, she will be visited by m ghosts tonight. Anya has lots of candles prepared for the visits, each candle can produce light for exactly t seconds. It takes the girl one second to light one candle. More formally, Anya can spend one second to light one candle, then this candle burns for exactly t seconds and then goes out and can no longer be used.

For each of the m ghosts Anya knows the time at which it comes: the i-th visit will happen w_i seconds after midnight, all w_i's are distinct. Each visit lasts exactly one second.

What is the minimum number of candles Anya should use so that during each visit, at least r candles are burning? Anya can start to light a candle at any time that is integer number of seconds from midnight, possibly, at the time before midnight. **That means, she can start to light a candle integer number of seconds before midnight or integer number of seconds after a midnight, or in other words in any integer moment of time.**

Input

The first line contains three integers m, t, r ($1 \le m, t, r \le 300$), representing the number of ghosts to visit Anya, the duration of a candle's burning and the minimum number of candles that should burn during each visit.

The next line contains m space-separated numbers w_i ($1 \le i \le m$, $1 \le w_i \le 300$), the i-th of them repesents at what second after the midnight the i-th ghost will come. All w_i's are distinct, they follow in the strictly increasing order.

Output

If it is possible to make at least r candles burn during each visit, then print the minimum number of candles that Anya needs to light for that.

If that is impossible, print -1.

Examples

input
1 8 3 10
output
3

input
2 10 1 5 8
output
1

input
1 1 3 10
output
-1

Note

Anya can start lighting a candle in the same second with ghost visit. But this candle isn't counted as burning at this visit.

It takes exactly one second to light up a candle and only after that second this candle is considered burning; it means that if Anya starts lighting candle at moment x, candle is buring from second x + 1 to second x + t inclusively.

In the first sample test three candles are enough. For example, Anya can start lighting them at the 3-rd, 5-th and 7-th seconds after the midnight.

In the second sample test one candle is enough. For example, Anya can start lighting it one second before the midnight.

In the third sample test the answer is -1, since during each second at most one candle can burn but Anya needs three candles to light up the room at the moment when the ghost comes.

D. Tanya and Password

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

While dad was at work, a little girl Tanya decided to play with dad's password to his secret database. Dad's password is a string consisting of $n + 2$ characters. She has written all the possible n three-letter continuous substrings of the password on pieces of paper, one for each piece of paper, and threw the password out. Each three-letter substring was written the number of times it occurred in the password. Thus, Tanya ended up with n pieces of paper.

Then Tanya realized that dad will be upset to learn about her game and decided to restore the password or at least any string corresponding to the final set of three-letter strings. You have to help her in this difficult task. We know that dad's password consisted of lowercase and uppercase letters of the Latin alphabet and digits. Uppercase and lowercase letters of the Latin alphabet are considered distinct.

Input

The first line contains integer n ($1 \le n \le 2 \cdot 10^5$), the number of three-letter substrings Tanya got.

Next n lines contain three letters each, forming the substring of dad's password. Each character in the input is a lowercase or uppercase Latin letter or a digit.

Output

If Tanya made a mistake somewhere during the game and the strings that correspond to the given set of substrings don't exist, print "NO".

If it is possible to restore the string that corresponds to given set of substrings, print "YES", and then print any suitable password option.

Examples

input
5 aca aba aba cab bac
output
YES abacaba

input
4 abc bCb cb1 b13
output
NO

input
7 aaa aaa aaa aaa aaa aaa aaa
output
YES aaaaaaaaa

E. Arthur and Brackets

time limit per test: 2 seconds
memory limit per test: 128 megabytes
input: standard input
output: standard output

Notice that the memory limit is non-standard.

Recently Arthur and Sasha have studied correct bracket sequences. Arthur understood this topic perfectly and become so amazed about correct bracket sequences, so he even got himself a favorite correct bracket sequence of length $2n$. Unlike Arthur, Sasha understood the topic very badly, and broke Arthur's favorite correct bracket sequence just to spite him.

All Arthur remembers about his favorite sequence is for each opening parenthesis ('(') the approximate distance to the corresponding closing one (')'). For the i-th opening bracket he remembers the segment $[l_i, r_i]$, containing the distance to the corresponding closing bracket.

Formally speaking, for the i-th opening bracket (in order from left to right) we know that the difference of its position and the position of the corresponding closing bracket belongs to the segment $[l_i, r_i]$.

Help Arthur restore his favorite correct bracket sequence!

Input

The first line contains integer n ($1 \le n \le 600$), the number of opening brackets in Arthur's favorite correct bracket sequence.

Next n lines contain numbers l_i and r_i ($1 \le l_i \le r_i < 2n$), representing the segment where lies the distance from the i-th opening bracket and the corresponding closing one.

The descriptions of the segments are given in the order in which the opening brackets occur in Arthur's favorite sequence if we list them from left to right.

Output

If it is possible to restore the correct bracket sequence by the given data, print any possible choice.

If Arthur got something wrong, and there are no sequences corresponding to the given information, print a single line "IMPOSSIBLE" (without the quotes).

Examples

input
4 1 1 1 1 1 1 1 1
output
()()()()

input
3 5 5 3 3 1 1
output
((()))

input
3 5 5 3 3 2 2
output
IMPOSSIBLE

input
3 2 3 1 4 1 4
output
(())()