



Examination 2

JOI-kun goes to IOI High School, where the final exam is held soon. In the exam, students will be tested on whether they can correctly calculate the value of an **IOI function**. An IOI function is a string obtained by one of the following six rules defined by IOI High School, which maps integers between 1 and 10^9 (inclusive) to boolean values, either True or False.

1. Let a be an integer between 1 and 10^9 (inclusive), then $[a]$ is an IOI function (a is the string representation of a in decimal notation). This IOI function maps integers greater than or equal to a to True, and integers less than a to False.
2. Let f be an IOI function, then (f) is also an IOI function. This IOI function maps the same integers to True and False as f does.
3. Let f be an IOI function, then $!f$ is also an IOI function. This IOI function maps integers that f maps to True to False, and vice versa.
4. Let f and g be IOI functions, then $f \& g$ is also an IOI function. This IOI function maps integers to True if both f and g map them to True, and to False otherwise.
5. Let f and g be IOI functions, then $f \wedge g$ is also an IOI function. This IOI function maps integers to True if exactly one of f or g maps them to True, and to False otherwise.
6. Let f and g be IOI functions, then $f | g$ is also an IOI function. This IOI function maps integers to True if at least one of f or g maps them to True, and to False otherwise.

If an IOI function is obtained using multiple rules, the rule with the higher number determines the boolean value that the IOI function maps integers to. For example, for $[1] \& [2] | [3]$, rule 6 is applied to $f = [1] \& [2]$ and $g = [3]$ (rather than applying rule 4 to $f = [1]$ and $g = [2] | [3]$). Additionally, for rules 4, 5, and 6, the rule is applied so that f becomes as long as possible. For example, for $[4] \wedge [5] \wedge [6]$, rule 5 is applied to $f = [4] \wedge [5]$ and $g = [6]$ (rather than applying rule 5 to $f = [4]$ and $g = [5] \wedge [6]$).

To prepare for the exam, JOI-kun has prepared an IOI function S of length N and intends to practice determining the boolean values that this IOI function maps Q integers X_1, X_2, \dots, X_Q to. He asks for your help, as you are proficient with handling IOI functions, to create a sample solution.

Write a program which, given N, Q, S and X_1, X_2, \dots, X_Q , determines the boolean values that the IOI function S maps integers X_1, X_2, \dots, X_Q to.



Input

The input is given from Standard Input in the following format:

N Q
 S
 X_1
 X_2
 \vdots
 X_Q

Output

Print Q lines to Standard Output. i -th line ($1 \leq i \leq Q$) should contain a single boolean value which the IOI function S maps the integer X_i to.

Constraints

- $1 \leq N \leq 1\,000\,000$.
- $1 \leq Q \leq 200\,000$.
- S is an IOI function of length N .
- $1 \leq X_i \leq 10^9$ ($1 \leq i \leq Q$).
- N , Q , and X_i ($1 \leq i \leq Q$) are integers.

Subtasks

1. (5 points) S does not contain & or |.
2. (20 points) $Q = 1$.
3. (10 points) $N \leq 10\,000$.
4. (6 points) S does not contain ! or ^.
5. (12 points) When rule 4 or rule 6 was applied in the process of obtaining S , at least one of f or g was an IOI function obtained from rule 1.
6. (20 points) $N \leq 400\,000$.
7. (27 points) No additional constraints.



Sample Input and Output

Sample Input 1	Sample Output 1
15 5	True
(![2] [3])&![4]	False
1	True
2	False
3	False
4	
5	

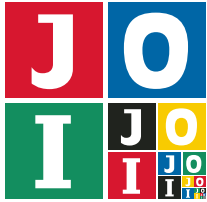
For some of the IOI functions that appear in the process of obtaining S according to the rules in the problem statement, the boolean values they map integers X_i ($1 \leq i \leq Q$) to are as shown in the following table.

X_i	![2]	[3]	![2] [3]	![4]	(![2] [3])&![4]
1	True	False	True	True	True
2	False	False	False	True	False
3	False	True	True	True	True
4	False	True	True	False	False
5	False	True	True	False	False

This sample input satisfies the constraints of subtasks 3, 6 and 7.

Sample Input 2	Sample Output 2
20 4	True
(!![23])^((([116])))	False
54	False
1	True
200	
89	

This sample input satisfies the constraints of subtasks 1, 3, 5, 6 and 7.



Sample Input3	Sample Output3
32 4	True
[2] [5]&[1] (([10000000000]) [7])	True
4	True
10	False
6	
1	

This sample input satisfies the constraints of subtasks 3, 4, 6 and 7.