

Video Service

You are the single engineer who has newly been placed in charge of all of the servers at Metflicks after the previous employee quit for 3% more total compensation at a rival streaming service. After coming to terms with the fact that you are doomed to manage the terabytes of data requested of your precious servers each day, you pull yourself up by your bootstraps and get to work. Once you finally figure out how everything works, you stumble upon a note scrawled in what looks like ketchup tucked away in your desk. It is a hastily written list titled DO NOT SERVE UNDER ANY CIRCUMSTANCES, with many IP addresses listed beneath. You find another list written in what smells like mustard titled BAD DATA, with very sketchy descriptions of packet payloads that when strung together, cause all of your servers to catch fire.



Write a program that, given a list of IP addresses and data sequences, rejects all packets that contain IP addresses that are not to be served and halts acknowledgements when multiple successive packets are combined to form bad data. Packets with lower sequence numbers contain higher significance bits.

In addition, known valid users can make bad data requests, putting them on the naughty list. Once a user has been marked as DO NOT SERVE, they cannot get off this list and are permanently barred from using Metflicks.

Input

The first line of input is a number $2 \leq a \leq 2^{20}$ of DO NOT SERVE IP addresses. The next a lines will contain 32-bit representations of IP addresses

The next line of input is a number $2 \leq b \leq 2^{11}$ of possible bitstrings of bad data. The next b lines will contain bitstrings of length 2^5 .

The following line of input is a number $2^2 \leq p \leq 2^{500}$, where you will receive p total packets. The next p lines will be packets in the form:

32-bit source IP address i		32-bit data bitstring d
---------------------------------	--	------------------------------

The remaining input is a number $2 \leq u \leq 2^{20}$ representing the number of user IP address to test. The following u lines of input are IP addresses that may or may not have sent a request to your servers.

Output

For each IP address u , print 1 to mark that this user only had good requests or print 0 to mark that this user either was originally on the DO NOT SERVE list or earned their way on by sending bad data. A user moves from being good to bad if they send three bad messages in the same chain. Print nothing if this user did not make a request.

Languages

This functions in java, python and C++, or pretty much any language that has support of integers over size 2^{500} . Runtime should not be a problem unless I/O is extremely slow.

Sample Input

```
3
00111111100101011011111111001000
00111111011100010110101011100110
00010010001100000110010001001000
4
00100000010010101011010110110010
01110010011111001011101100011110
01110101001011011001011111001001
0001101011111101110100011000000
10
0110011111100110011001010010100000011010111111101110100011000000
0110011111100110011001010010100001110101001011011001011111001001
0110011111100110011001010010100000110001110101010110001010100101
0110011111100110011001010010100000010011100111011000010101101101
011001111110011001100101001010000101100101010110100111111100101
0110011111100110011001010010100001110001011011111110111000111001
0110011111100110011001010010100001111001101111100010001111001110
0110011111100110011001010010100000001101100011000000000010100110
0110011111100110011001010010100001111011010110010111101111001011
0110011111100110011001010010100001111100110111010001001011100111
6
01100111111001100110010100101000
00011010011010010000010101001010
01011001111111101100010100010010
00010010001100000110010001001000
01011000110000010110011001000010
00111111011100010110101011100110
```

Sample Output

```
100
```