NOI '21 P4 - Quantum Communication

Time limit: 3.0s **Memory limit:** 384M

Alice and Bob are engaging in quantum communication. Their language of communication is a dictionary S of size n. In the dictionary, a word s_i $(1 \leq i \leq n)$ may be represented by a 256-bit binary string. In this problem, s_i may be generated by calling function `gen`. The contestants may refer to gen.cpp, and the parameters `n,a1,a2` will be specified by the test cases.

Alice and Bob will communicate for m rounds next. In each round, Alice will send bob *exactly one* word in the dictionary. However, their communication channel is not reliable and may be disturbed by noise. More formally, for the i-th round of communication, suppose the word Alice wants to send is x_i, then at most k_i bits of the binary string will flip. In other words, suppose the binary string Bob receives is y_i, then y_i is different from x_i for at most k_i positions. y_i does not have to be in dictionary S.

At the same time, Bob knows Eve has invaded the communication channel and is going to manipulate the communication between Alice and Bob. Eve will replace the binary string Bob is going into receive with an arbitrary 256-bit binary string, and the arbitrary 256-bit binary string does not have to be in the dictionary. Eve does not have to manipulate all rounds of communication.

Now Bob asks you for help, and you need to determine whether it is possible that a round of communication is *not* manipulated by Eve given the string Bob received and the threshold for noise disturbance k_i $(0 \leq k_i \leq 15)$. In other words, you need to check whether the binary string Bob received can be obtained by flipping at most k_i bits of a word in the dictionary. If it is possible that the round of communication is not manipulated by Eve, output 1. Otherwise, output 0. Bob trusts your ability, so you need to answer the queries *online*. For the specifics, see the input section.

To save the time spent on I/O, the strings Bob received are given by 64-bit hexadecimal strings. The hexadecimal strings contain numerals 0-9 and upper-case English letters A-F. A-F represents 10-15 in that order. A hexadecimal string may be converted to a binary string bit by bit. For example, `5` will correspond to `0101`, `A` will correspond to `1010`, and `C` will correspond to `1100`.

Input Specification

The first line of the input contains four nonnegative integers n, m, a_1, a_2 denoting the size of the dictionary, the number of rounds of communication, and the initial values of parameters `a1` and `a2` for the function `gen`. Contestants need to generate the dictionary using the `gen` function mentioned in the problem statement. Contestants may copy and use the code in `gen.cpp`. The Boolean array `s[N+1][256]` is just the list of all words.

For the next m lines, each line contains a hexadecimal string of length 64 and a nonnegative integer k_i denoting the final binary string Bob received in round i and the threshold for noise disturbance.

To make sure contestants are answering the queries online, after the contestants recover the 256-bit binary string based on the hexadecimal string, contestants need to perform XOR operation with `lastans` bitwise to recover the real binary string Bob received in the round. `lastans` $\in \{0, 1\}$ denotes the answer to the query last round. Before the first round, the initial value of `lastans` is 0.

Notice: when reading or writing `unsigned long long` variables using `scanf` or `printf`, use `llu`.

Output Specification

The output consists of m lines. Each line contains an integer 0 or 1 denoting the answer to the query.

Sample

For convenience, we will give the words in the dictionary explicitly, reduce the word length to 4, and allow offline answers to the queries.

Suppose the dictionary size is $n = 2$. The words are 1010 and 0111.

For query `B = 1011` and $k_1 = 1$, the output shall be 1 since `B` may be obtained by flipping the 4-th bit of 1010.

For query `B = 0001` and $k_2 = 2$, the output shall be 1 since 0001 may be obtained by flipping the second and the third bit of 0111.

For query `B = 0001` and $k_3 = 1$, the output shall be 0.

The remaining samples can be found here.

Constraints

For all test cases, $1 \le n \le 4 \times 10^5$, $1 \le m \le 1.2 \times 10^5$, $0 \le k \le 15$, and a_1, a_2 are uniformly random among $[0, 2^{64} - 1]$.

Test case	$n =$	$m =$	$k_i \le$	Additional Constraints

1	10		2	None.
2	500		15	
3	$1\,000$		0	
4	$2\,000$		2	
5	$5\,000$		15	
6	$10\,000$			
7	$20\,000$			
8	$100\,000$		1	
9	$400\,000$	$120\,000$		
10	$50\,000$		2	
11	$70\,000$		3	
12	$100\,000$		2	
13	$30\,000$		5	
14	$60\,000$		4	
15	$120\,000$		5	
16	$60\,000$		8	The query strings are generated randomly.
17	$120\,000$		12	
18	$400\,000$	$100\,000$	15	
19	$30\,000$		7	None.
20	$60\,000$		9	
21	$90\,000$		11	
22	$200\,000$	$120\,000$	12	
23	$400\,000$	$80\,000$	15	
24	$400\,000$	$100\,000$	15	
25	$400\,000$	$120\,000$	15	