

第 1 题：寻宝游戏(hunt)，运行时限 1s，内存上限 512M，100 分。

【问题描述】

某大学每年都会有一次 Mystery Hunt 的活动，玩家需要根据设置的线索解谜，找到宝藏的位置，前一年获胜的队伍可以获得这一年出题的机会。

作为新生的你，对这个活动非常感兴趣。你每天都要从西向东经过教学楼一条很长的走廊，这条走廊是如此的长，以至于它被人戏称为 infinite corridor。一次，你经过这条走廊的时候，注意到在走廊的墙壁上隐藏着 n 个等长的二进制的数字，长度均为 m 。你从西向东将这些数字记录了下来，形成一个含有 n 个数的二进制数组 $a_1, a_2, \dots, a_n$ 。

很快，在最新的一期 Voo Doo 杂志上，你发现了 q 个长度也为 m 的二进制串 $r_1, r_2, \dots, r_q$ 。

聪明的你很快发现了这些数字的含义。

保持数组 $a_1, a_2, \dots, a_n$ 的元素顺序不变，你可以在它们之间插入 $\wedge$ （按位与运算）或者 $\vee$ （按位或运算）两种二进制运算符。例如： $11011 \wedge 00111 = 00011$ ， $11011 \vee 00111 = 11111$ 。

你需要插入恰好 n 个运算符，相邻两个数之间恰好一个，在第一个数的左边还有一个。如果我们在第一个运算符的左边补入一个 0，这就形成了一个运算式，我们可以计算它的值。与往常一样，运算顺序是从左往右。有趣的是，出题人已经告诉你这个值的可能的集合——Voo Doo 杂志里的那些二进制数 $r_1, r_2, \dots, r_q$ ，而解谜的方法，就是对 $r_1, r_2, \dots, r_q$ 中的每一个值 r_i ，分别计算出有多少种方法填入这 n 个运算符，使得这个运算式的值是 r_i 。

然而，infinite corridor 真的很长，这意味着数据范围可能非常大。因此，答案也可能非常大，但是你发现由于谜题的特殊性，你只要求答案模 $1000000007(10^9 + 7, \text{一个质数})$ 的值。

【程序文件名】

源程序文件名为 hunt.cpp/c/pas。

【输入格式】

输入文件名为 hunt.in。

第一行三个数 n, m, q ，含义如题所述。

接下来 n 行，其中第 i 行有一个长度为 m 的二进制串，左边是最高位，表示 a_i 。

接下来 q 行，其中第 i 行有一个长度为 m 的二进制串，左边是最高位，表示 r_i 。

【输出格式】

输出文件名为 hunt.out。

输出 q 行，每行一个数，其中第 i 行表示对应于 r_i 的答案。

【输入输出样例 1】

hunt.in	hunt.out
5 5 1 01110 11011 10000 01010 00100 00100	6

【样例解释 1】

有以下且仅有以下六个运算式的值是 00100_2 ：(下标 2 表示被标识的数是二进制数)

$$\begin{aligned} &0 \wedge 01110_2 \wedge 11011_2 \vee 10000_2 \wedge 01010_2 \vee 00100_2 \\ &0 \vee 01110_2 \vee 11011_2 \wedge 10000_2 \wedge 01010_2 \vee 00100_2 \\ &0 \wedge 01110_2 \vee 11011_2 \wedge 10000_2 \wedge 01010_2 \vee 00100_2 \\ &0 \vee 01110_2 \wedge 11011_2 \wedge 10000_2 \wedge 01010_2 \vee 00100_2 \\ &0 \wedge 01110_2 \wedge 11011_2 \wedge 10000_2 \wedge 01010_2 \vee 00100_2 \\ &0 \vee 01110_2 \vee 11011_2 \vee 10000_2 \vee 01010_2 \wedge 00100_2 \end{aligned}$$

【输入输出样例 2】

hunt.in	hunt.out
10 10 3	69
0100011011	0
0110100101	5
1100010100	
0111000110	
1100011110	
0001110100	
0001101110	
0110100001	
1110001010	
0010011101	
0110011111	
1101001010	
0010001001	

【数据范围】

- 对于 10%的数据， $n \leq 20, m \leq 30, q = 1$
- 对于另外 20%的数据， $n \leq 1000, m \leq 16$
- 对于另外 40%的数据， $n \leq 500, m \leq 1000$
- 对于 100%的数据， $1 \leq n \leq 1000, 1 \leq m \leq 5000, 1 \leq q \leq 1000$

【提示】

输入文件可能很大，请注意读入效率。

【编译命令】

- 对于 c++ 语言： `g++ -o hunt hunt.cpp -lm`
- 对于 c 语言： `gcc -o hunt hunt.c -lm`
- 对于 pascal 语言： `fpc hunt.pas`