

## J. Now or Never / Leximin

时间限制：2.0 秒

空间限制：512 MiB

## 【题目描述】

对于一个长度为  $l$  的 01 串  $w = w_1w_2\dots w_l$ ，定义其支撑序列  $\text{supp}(w)$  为  $[1, 2, \dots, l]$  的一个子序列，其中  $i \in \text{supp}(w)$  当且仅当  $w_i = 1$ 。例如， $\text{supp}(001100110) = [3, 4, 7, 8]$ 。特别地，全零串的支撑序列为空序列  $\varepsilon$ 。

给定一个 01 串集合  $S$ ，其中包含  $n$  个长度为  $m$  的 01 串  $s_1, s_2, \dots, s_n$ 。再给定  $q$  组询问，第  $i$  组询问给出一个长度为  $m$  的 01 串  $t_i$ 。你需要输出一个长度为  $m$  的 01 串  $u_i$  满足以下条件：

1. 存在一个  $S$  的子集  $T$ ，其中  $T$  可以为空也可以为  $S$  本身，满足  $u_i = t_i \oplus (\bigoplus_{v \in T} v)$ ，其中  $\oplus$  表示按位异或操作，即  $u_i$  为  $t_i$  与  $T$  中所有 01 串的异或和；
2. 在以上条件的基础之上， $u_i$  的支撑序列的字典序尽可能小。

对于两个序列  $A, B$ ，称  $A$  的字典序小于  $B$ ，当且仅当  $A$  是  $B$  的一个真前缀，或者  $A$  和  $B$  在第一个相异的位置  $p$  上满足  $A_p < B_p$ 。

## 【输入格式】

从标准输入读入数据。

输入的第一行三个正整数  $n, m, q$  ( $1 \leq n, m, q \leq 2000$ )，分别表示集合  $S$  的大小、01 串的长度以及询问组数。

接下来  $n$  行，第  $i$  行一个长度为  $m$  的 01 串  $s_i$ 。

最后  $q$  行，第  $i$  行一个长度为  $m$  的 01 串  $t_i$  描述一组询问。

## 【输出格式】

输出到标准输出。

对于第  $i$  组询问输出一行表示满足题设条件的长度为  $m$  的 01 串  $u_i$ 。

## 【样例 1 输入】

```
1 1 3 2
2 110
3 010
4 101
```

**【样例 1 输出】**

```
1 100
2 101
```

**【样例 1 解释】**

对于第一组测试数据，满足第一个条件的串有 **010** 和 **100**。二者支撑序列分别为  $\text{supp}(010) = [2]$ ， $\text{supp}(100) = [1]$ ，其中字典序更小的是  $[1]$ 。

对于第二组测试数据，满足第一个条件的串有 **101** 和 **011**。二者支撑序列分别为  $\text{supp}(101) = [1, 3]$ ， $\text{supp}(011) = [2, 3]$ ，其中字典序更小的是  $[1, 3]$ 。

**【样例 2 输入】**

```
1 2 4 4
2 1100
3 1010
4 1000
5 0001
6 0110
7 0011
```

**【样例 2 输出】**

```
1 1000
2 1101
3 0000
4 1111
```

**【样例 2 解释】**

对于第一组测试数据，满足第一个条件的串有 **1000**、**0100**、**0010** 和 **1110**，其中字典序最小的支撑序列是  $\text{supp}(1000) = [1]$ 。

对于第二组测试数据，满足第一个条件的串有 **0001**、**1101**、**1011** 和 **0111**，其中字典序最小的支撑序列是  $\text{supp}(1101) = [1, 2, 4]$ 。

对于第三组测试数据，满足第一个条件的串有 **0110**、**1010**、**1100** 和 **0000**，其中字典序最小的支撑序列是  $\text{supp}(0000) = \varepsilon$ ，也即空序列。

对于第四组测试数据，满足第一个条件的串有0011、1111、1001 和 0101，其中字典序最小的支撑序列是  $\text{supp}(1111) = [1, 2, 3, 4]$ 。

**【样例 3 输入】**

```
1 3 9 7
2 011001111
3 101110001
4 110010100
5 010110110
6 101010100
7 000101101
8 001011100
9 100011111
10 100111000
11 001000101
```

**【样例 3 输出】**

```
1 111101101
2 110110001
3 110111001
4 111100010
5 111111010
6 111110111
7 111111011
```

**【样例 4 输入】**

```
1 9 24 9
2 100001011101000000000000
3 100011001100100001000000
4 010101000010100111110100
5 101110000010101110110010
6 100110110010011100000000
7 111111000010100101111011
8 000010110001001011010101
```

```
9 010101100111000010100111
10 111001111111000000000000
11 111000110110110000000000
12 011100101000100001000000
13 101001101000111011001100
14 1000111000111110001000000
15 1000010010110000000000000
16 1011101100011111000000000
17 100011100101100010110000
18 101001001100101011000000
19 100101100110100111000000
```

#### 【样例 4 输出】

```
1 111111111100001001010011
2 111111101111001101010101
3 111111101100001011000101
4 111111101101110101111011
5 111111111111000010111011
6 111111111111110101011010
7 111111101110101001001101
8 111111101101111110011010
9 111111111110100001000000
```

#### 【提示】

题目名称是什么意思？