

Problem H. Power of three

Input file: **stdin**
Output file: **stdout**
Time limit: 1 second
Memory limit: 512 megabytes

bobo has a binary matrix of size $n \times m$. Today bobo is going to remove some (maybe none) of rows to maximize the *boboness* of the matrix.

The *boboness* is defined as follows. If there are odd number of ones in the i -th column, $a_i \cdot 3^{b_i}$ points are added up to the *boboness*, initially 0.

Find the maximum of *boboness*.

Input

The first line contains 2 integers n, m ($1 \leq n \leq 200000, 1 \leq m \leq 70$).

Each of the following n lines contains m integers which denotes the matrix.

Each of the last m lines contains 2 integers a_i, b_i ($a_i \in \{-1, 1\}, 1 \leq b_i \leq 35$).

It is guaranteed that for all $i \neq j$, either $a_i \neq a_j$ or $b_i \neq b_j$.

Output

A single integer denotes the maximum of *boboness*.

Sample input and output

stdin	stdout
2 4 1101 0010 -1 1 -1 2 1 1 1 2	3
1 1 1 -1 1	0