

E. Lucky Arrays

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Maxim loves interesting problems. He decided to share one such problem with you.

Initially there is an array a , consisting of n zeroes. The elements of the array are indexed, starting from 1. Then follow queries to change array a . Each query is characterized by two integers v_i, t_i . In the answer to the query we should make the v_i -th array element equal t_i ($a_{v_i} = t_i$; $1 \leq v_i \leq n$).

Maxim thinks that some pairs of integers (x, y) are good and some are not. Maxim thinks that array a , consisting of n integers, is lucky, if for all integer i , ($1 \leq i \leq n - 1$) the pair of integers (a_i, a_{i+1}) — is good. Note that the order of numbers in the pairs is important, that is, specifically, $(1, 2) \neq (2, 1)$.

After each query to change array a Maxim wants to know, how many ways there are to replace all zeroes in array a with integers from one to three so as to make the resulting array (without zeroes) lucky. Of course, distinct zeroes can be replaced by distinct integers.

Maxim told you the sequence of queries and all pairs of integers he considers lucky. Help Maxim, solve this problem for him.

Input

The first line contains integers n and m ($1 \leq n, m \leq 77777$) — the number of elements in the array and the number of commands.

The next three lines contain matrix w , consisting only of zeroes and ones; the j -th number in the i -th of these lines — $w_{i,j}$. If $w_{i,j} = 1$ ($1 \leq i, j \leq 3$), then pair (i, j) is good, otherwise it is not good. Matrix does not have to be symmetric relative to the main diagonal.

Next m lines contain pairs of integers v_i, t_i ($1 \leq v_i \leq n, 0 \leq t_i \leq 3$) — the queries to change the array.

Output

Print m integers — the i -th number should equal to the number of ways to replace all zeroes in array a (changed after the i -th query) by integers from one to three so as to make the resulting array (without zeroes) lucky. Separate the numbers by whitespaces. As the answers can be rather large, print the remainder from dividing them by 777777777.

Examples

input
3 10 1 1 0 1 0 0 1 1 1 1 1 1 3 2 2 3 0 2 1 3 0 3 1 2 0 3 1 1 0
output
3 6

1
1
2
2
1
3
3
6