

Edit Distance Parity

Input file: **standard input**
Output file: **standard output**
Time limit: **3 seconds**
Memory limit: **256 megabytes**

Busy Beaver has a sequence of N integers a_1, a_2, \dots, a_N and a sequence of M integers b_1, b_2, \dots, b_M .

For every $1 \leq i \leq N$ and $1 \leq j \leq M$, he computes the edit distance between the sequences a_1, a_2, \dots, a_i and b_1, b_2, \dots, b_j . The edit distance between two sequences is defined as the minimum number of operations needed to go from the first sequence to the second sequence if the following three operations are allowed:

- **Insertions:** Any integer can be inserted in any position in the sequence. The rest of the sequence remains the same. For example, in one operation, the sequence 1, 2, 3, 4 can become 1, 2, 5, 3, 4.
- **Deletions:** Any integer present in the sequence can be deleted. The rest of the sequence remains the same. For example, in one operation, the sequence 1, 2, 3, 4 can become 1, 2, 4.
- **Replacements:** Any integer present in the sequence can be replaced by another integer. The rest of the sequence remains the same. For example, in one operation, the sequence 1, 2, 3, 4 can become 1, 2, 5, 4.

As storing each of these values takes too much space, Busy Beaver only stores the least significant bit of each edit distance as $d_{i,j}$ (0 if it is even and 1 if it is odd).

Unfortunately, Busy Beaver has lost the original sequences and only has the values $d_{i,j}$ for each i and j . Help him find any pair of sequences a_1, a_2, \dots, a_N and b_1, b_2, \dots, b_M that will give the same values for $d_{i,j}$. It is possible that no original sequences can produce $d_{i,j}$, in which case you should report this.

Input

The first line contains a single integer T ($1 \leq T \leq 2 \cdot 10^3$) — the number of test cases.

The first line of each test case contains two integers N and M ($1 \leq N, M \leq 2 \cdot 10^3$).

The i -th of the next N lines of each test case contains M integers $d_{i,1}, d_{i,2}, \dots, d_{i,M}$ ($0 \leq d_{i,j} \leq 1$).

The sum of N across all test cases does not exceed $2 \cdot 10^3$.

The sum of M across all test cases does not exceed $2 \cdot 10^3$.

Output

For each test case, print a sequence of N integers a_1, a_2, \dots, a_N ($0 \leq a_i \leq 10^9$) on the first line, and a sequence of M integers b_1, b_2, \dots, b_M ($0 \leq b_i \leq 10^9$) on the second line, such that all $d_{i,j}$ have the same parity as the edit distance between a_1, a_2, \dots, a_i and b_1, b_2, \dots, b_j .

If there are no such sequences, print -1 on a single line.

It can be shown that if there are any sequences a_1, a_2, \dots, a_N and b_1, b_2, \dots, b_M that work, then there are sequences that satisfy $0 \leq a_i \leq 10^9$ and $0 \leq b_i \leq 10^9$.

Example

standard input	standard output
2	0 1 2 3 3
5 3	3 3 1
1 0 1	-1
0 0 0	
1 1 1	
1 1 0	
0 1 0	
4 6	
1 1 0 1 0 1	
1 1 1 0 1 0	
0 1 1 1 0 1	
1 0 0 0 0 0	

Note

In the first test case, it can be checked that $[0, 1, 2, 3, 3]$ and $[3, 3, 1]$ work. For example, the edit distance for $[0, 1, 2, 3]$ and $[3, 3]$ is 3 because the following operations work:

1. Delete the first element of $[0, 1, 2, 3]$ to get $[1, 2, 3]$
2. Delete the first element of $[1, 2, 3]$ to get $[2, 3]$
3. Replace the first element of $[2, 3]$ to get $[3, 3]$

The least significant bit of 3 is 1, which matches the input.

In the second test case, it can be shown that there do not exist any valid sequences.