7052   Memory

Little Bob's computer has 2^n bytes of memory. For convenience, n-bit integers 0 to $2^n - 1$ are used to index these bytes.

Now he wants to assign a value to each byte of the memory. In this problem, a byte is composed of m bits. Therefore he can only assign 0 to $2^m - 1$ (inclusive) to a single byte.

Bob has some preferences on which value to be assigned to each byte. For the byte indexed by i, if it is assigned with value j ($0 \leq j < 2^m$), the preference score for it is $w_{i,j}$.

In addition, each byte has a threshold value. For two different bytes indexed by a and b, if the following two conditions are satisfied, there will be an additional score (u_a xor u_b):

1. a and b only have one bit of difference in their binary forms;

2. The assigned value of byte a is not less than its threshold value, or the assigned value of byte b is not less than its threshold value.

The total score of an assignment solution is the sum of the preference scores of all bytes plus the sum of all additional scores.

Bob wants to find an assignment solution with the maximum total score. If there are multiple solutions, you can output any of them.

Input

The first line contains an integer T ($T \leq 3$), denoting the number of the test cases.

For each test case, the first line contains two integers, n and m ($1 \leq n, m \leq 8$), as mentioned above. The second line contains 2^n integers, and the i-th integer is the threshold value for byte i. The threshold values are between 0 and $2^m - 1$, inclusively. The third line contains 2^n integers, and the i-th integer is u_i ($0 \leq u_i < 1024$). The next 2^n lines give all preference scores. Each line contains 2^m integers, and the j-th integer of the i-th line is $w_{i,j}$ ($-1024 \leq w_{i_j} < 1024$).

Output

For each test case, output one line consisting of 2^n integers between 0 and $2^m - 1$, and the i-th integer is the value assigned to byte i in the assignment solution with the maximum total score.

Sample Input

```
1
3 2
0 1 1 3 3 0 3 3
4 8 8 7 0 9 2 9
-9 -8 3 2
-9 -6 4 1
-6 -8 -5 3
```

```
3 -1 -4 -1
-6 -5 1 10
-10 7 3 -10
-3 -10 -4 -5
-2 -1 -9 1
```

Sample Output

```
2 2 3 0 3 1 0 3
```