

Fenwick Tree

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

Let $\text{lowbit}(x)$ denote the value of the lowest binary bit of x , e.g. $\text{lowbit}(12) = 4$, $\text{lowbit}(8) = 8$.

For an array a of length n , if an array s of length n satisfies $s_k = \left(\sum_{i=k-\text{lowbit}(k)+1}^k a_i \right) \bmod 998\,244\,353$ for all k , then s is called the *Fenwick Tree* of a . Let's denote it as $s = f(a)$.

For a positive integer k and an array a , $f^k(a)$ is defined as follows:

$$f^k(a) = \begin{cases} f(a) & \text{if } k = 1 \\ f(f^{k-1}(a)) & \text{otherwise.} \end{cases}$$

You are given an array b of length n and a positive integer k . Find an array a that satisfies $0 \leq a_i < 998\,244\,353$ and $f^k(a) = b$. It can be proved that an answer always exists. If there are multiple possible answers, you may print any of them.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains two positive integers n ($1 \leq n \leq 2 \cdot 10^5$) and k ($1 \leq k \leq 10^9$), representing the length of the array and the number of times the function f is performed.

The second line of each test case contains an array b_1, b_2, \dots, b_n ($0 \leq b_i < 998\,244\,353$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single line, containing a valid array a of length n .

Example

standard input	standard output
2	1 1 1 1 1 1 1 1
8 1	1 2 3 4 5 6
1 2 1 4 1 2 1 8	
6 2	
1 4 3 17 5 16	

Note

In the first test case, it can be seen that $f^1([1, 1, 1, 1, 1, 1, 1, 1]) = [1, 2, 1, 4, 1, 2, 1, 8]$.

In the second test case, it can be seen that $f^2([1, 2, 3, 4, 5, 6]) = f^1([1, 3, 3, 10, 5, 11]) = [1, 4, 3, 17, 5, 16]$.