Problem H. Power of Two

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

$$2^{2^{2^{2^{2^{2^{2^{2^{2^{2}}}}}}}}}$$

SolarPea likes blowing up PolarSea's blog by sending power tower of 2. As the tower is too high, the stack of the web page overflows. So the blog no longer works.

Now SolarPea has n powers of two $a_1, a_2, \ldots, a_n$, x bitwise AND operators, y bitwise OR operators and z bitwise XOR operators. It is guaranteed that $n = x + y + z$.

Solarpea wants to construct an arithmetic expression with these numbers and operators. Formally, define $x_0 = 0$ and $x_i = x_{i-1}\ \mathrm{op}_i\ b_i$, where b is a permutation of a, which means we can rearrange a to get b, and op_i is one of the three types of bitwise operators above. Then x_n is the result of the expresstion.

The larger the expression, the more likely it is to make PolarSea's blog unable to work. SolarPea wants you to help him to find the largest x_n and construct such an expression. If there are multiple solutions, output any of them.

You need to process T test cases independently.

Input

The first line contains a single integer T $(1 \le T \le 10^5)$, denoting the number of test cases.

For each test case, the first line contains four integers n, x, y and z $(0 \le x, y, z \le n \le 65\,536, n = x+y+z)$. The next line contains n integers $c_1, c_2, \ldots, c_n$ $(0 \le c_i < n)$, where $a_i = 2^{c_i}$.

It is guaranteed that the sum of n over all test cases is no more than $1\,048\,576$.

Output

For each test case, output three lines.

The first line contains a 01-string of length n, representing the binary form of the largest x_n.

The next line contains a single 1-indexed string op of length n, where op_i represents the i-th operator. Here, we denote AND as & (ASCII 38), OR as | (ASCII 124), and XOR as ^ (ASCII 94). You should guarantee that there is exactly x AND operators, y OR operators and z XOR operators.

The third line contains n integers $d_1, d_2, \ldots, d_n$, the i-th of which representing the logarithm of b_i with base 2. That is, d is a permutaion of c.

If there are multiple solutions, output any of them.

Example

standard input	standard output
4 4 3 0 1 1 0 1 0 4 1 0 3 1 0 1 0 8 0 2 6 1 5 5 7 1 5 5 7 8 0 0 8 1 5 5 7 1 5 5 7	0010 &&^& 0 0 1 1 0011 ^^&^ 0 1 0 1 10100000 ^^|^^^^| 1 5 5 7 1 5 5 7 00000000 ^^^^^^^^ 1 5 5 7 1 5 5 7

Note