Base Conversion (Easy)

We want to make some base-conversion experiments. Here you can try basic methods.

Input

The first line of input contains three integers ***T***, the number of test cases, ***B1***, the first base, ***B2***, the second base.
Follow 2×***T*** lines.
For each test case, on the first line your are given one integer ***k***.
On the second line you are given ***k*** integers : the digits of ***N*** in base ***B1***.

$$N = a_0 \times B1^0 + ... + a_i \times B1^i + ... + a_{k-1} \times B1^{k-1}$$

Output

For each test case, you have to print the number ***N*** in base ***B2***. See sample for details.

Example

Input:
1 10 100
5
5 4 3 2 1

Output:
3 <--- Don't forget the length of N in base B2 ;-)
45 23 1

Explanations

For the lonely case, $N = 5\times10^0 + 4\times10^1 + 3\times10^2 + 2\times10^3 + 1\times10^4 = 12345$.
We have: $N = 45\times100^0 + 23\times100^1 + 1\times100^2$. You have to print 3, the number of digits, then the digits: 45, 23 and 1.

Constraints

$0 < T <= 200$
$1 < B1,B2 <= 10^9$
$1 < k <= 1000$
$0 <= a_i < B1$, $a_{k-1}>0$

If you find the constraints too easy, then you should try BASECONV.
The basic solution should give AC in 1.56s with Python3.
Have fun ;-)