# Base Conversion

Leo didn't do all the job in his last problem, somebody gave him the numbers in a convenient base. It was the bottleneck of the problem... Now your task is to do this job.

## 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
45 23 1

## Explanations

For the lonely case, $N = 5 \times 10^0 + 4 \times 10^1 + 3 \times 10^2 + 2 \times 10^3 + 1 \times 10^4 = 12345$.
We have: $N = 45 \times 100^0 + 23 \times 100^1 + 1 \times 100^2$. You have to print 3, the number of digits, then the digits: 45, 23 and 1.

## Constraints

$0 < T <= 50$
$1 < B1,B2 <= 10^9$
$1 < k <= 10000$
$0 <= a_i < B1$ , $a_{k-1} > 0$

Time limit is sqrt(T_basic_pike_code * T_awaited_python_code) = sqrt(13.34*6.97), based on my Python3/Pike experiments.
You may try before the tutorial edition.
**Have fun ;-)**

Edit(2017-02-11) : With compiler updates, a new time limit is set.
Time limit is sqrt(T_basic_pike_code * T_awaited_python_code) = sqrt(3.93*1.57), based on my

Python3/Pike experiments.
Thanks @Blue_Mary for pointing this out.