# 2227 Error Correcting Polynomials

Error correction is extremely useful in recovering from corrupted data. The abstract setting for this problem is: given a set of $n$ data elements such that at least $t$ of the data elements are uncorrupted, recover the complete uncorrupted data from it. To be able to do such a correction, data elements need to be of special form. Such forms are called *Error-correcting codes*. An error-correcting code that is widely used is as follows: the data elements are evaluations of a univariate polynomial of degree $d$ modulo a prime number $p$. The error-correction algorithm for this code was very complex until Madhu Sudan discovered a nice way of doing this. His algorithm is: given a set of $n$ points $y_1, \ldots, y_n$ (each value is between 0 and $p-1$, and also $n < p$) with at least $t$ of the $y$'s correct (this means that for the correct $y_j$, $y_j = P(j)$ for some unknown degree $d$ polynomial modulo $p$) do the following:

1. Construct a *non-zero* polynomial in two variables $Q(x, y)$ with the degree of $y$ being $e$ and the degree of $x$ being $f$ and $(e+1) * (f+1) > n$ and $e * d + f < t$ for which $Q(j, y_j) = 0$ for every $j$ between 1 and $n$. The exact values of $e$ and $f$ are determined using $t$, $d$, and $n$.

2. Factor the polynomial $Q$ into irreducible factors.

3. Consider all the factors of the form $(y - g(x))$ such that degree of $g$ is $d$ and identify one for which $y_j = g(j)$ for at least $t$ values.

This identified polynomial, $g$, is the desired one. Once the polynomial is identified, all the correct values of $y$ can be recovered by computing $g(j)$ for every $j$.

Madhu Sudan has written programs for factoring the two variable polynomial and for identifying the right polynomial as well as computed the right values of $e$ and $f$. He now just needs a program that constructs the two variable polynomial $Q$. Please help him in writing this program.

## Input

The number of test cases $T$ is written in the first line of the input.

For each test case, the first line contains the value of $n$, $p$, $e$, and $f$ (all these are between 1 and 100 satisfying the constraints mentioned above; in particular, $p$ is always a prime number) in that order. The $j$-th of the next $n$ lines contains the value $y_j$.

## Output

For each test case, output all the coefficients of the polynomial $Q$ in the following order: first output all the coefficients of the terms in which the degree of $y$ is zero and order these coefficients by increasing degree of $x$; then output all the coefficients of the terms in which degree of $y$ is one ordering these in the same way as before, and so on. Each coefficient must be on a separate line. If there are more than one solutions, output any one.

The output of different test cases must be separated by a blank line.

## Sample Input

```
2
3 7 1 1
6
```

```
4
5
4 11 1 2
2
3
0
0
```

## Sample Output

```
0
4
3
1
1
4
1
2
6
0
```