Problem P: Plus one

You have just realized that the average of all the numbers in the world is too small. Therefore, you decided to increase some of them inconspicuously.

Problem specification

You are given several integers. Increment each of them by one.

Input specification

The input consists of several lines, each line contains single integer x.
The absolute value of each integer lies between 0 and 1 000, inclusive.

Easy subproblem P1: There are exactly 50 lines of input.
Hard subproblem P2: There are exactly 1 000 lines of input.

Output specification

For each integer x in the input, output a single line with the integer $x + 1$.
Use precisely the same format that was used for x.

Example

input	output
1 46 41	2 47 42

Note: This example has only 3 lines of input.
The input files `p1.in` and `p2.in` contain 50 and 1 000 lines, respectively.

Also note: Please do NOT submit any programs.
For each subproblem, just produce and submit a correct output file.

Problem Q: Quite the cheater!

Your physics lab report is due tomorrow. However, you had no time to do the required experiments, as you spent all your time practicing for the IPSC. Therefore you decided to write a fake report quickly. Here is how to get a good grade for your lab report:

- It has to contain a lot of measurements.
- You already know the correct value you were supposed to measure. The mean of all "measured" values in your report has to be equal to that value.
- The values must look sufficiently random to avoid suspicion that you made them all up. (Yeah, right.) More formally, they must have a sufficient variance.

Problem specification

You are given two integers: the desired mean μ and the desired variance v.

Pick a number of measurements n and the values of those measurements $a_1, \ldots, a_n$ such that the mean of those values is exactly μ and their variance is (easy subproblem: at least v / hard subproblem: exactly v). Formally, your values must satisfy the following conditions:

- $10 \leq n \leq 1000$
- Each a_i is an integer between -10^9 and 10^9, inclusive.
- The value μ is exactly the mean: $\mu = (a_1 + \cdots + a_n)/n$.
- The variance of your values[1] is computed as follows: $(1/n) \cdot \Big((a_1 - \mu)^2 + \cdots + (a_n - \mu)^2\Big)$.
- In the easy subproblem Q1: the variance of your values must be at least v.
- In the hard subproblem Q2: the variance of your values must be exactly v.

Input specification

The first line of the input file contains an integer t specifying the number of test cases. Each test case is preceded by a blank line.

Each test case contains a single line with two integers: μ and v.

You may assume that $t \leq 100$, $|\mu| \leq 10^6$, and $0 \leq v \leq 10^9$.

Output specification

For each test case, output two lines. The first line should contain the number of values n, the second line a space-separated list of values $a_1, \ldots, a_n$. Any valid solution will be accepted.

Example

input

```
1

47 2080
```

output

```
11
34 -7 102 117 16 8 0 130 36 34 47
```

This would be a correct solution to both subproblems. I.e., this sequence of 11 values has mean exactly 47 and variance exactly 2080.

[1] If you are a statistics buff, note that we are not using the unbiased sample variance formula (the one with $1/(n-1)$ instead of $1/n$), as in our case the mean is known a priori. If the previous sentence makes no sense, just ignore it and use the formula in the problem statement.

Problem R: Rearranged alphabet

The string `abcabac` has a peculiar property: each permutation of `a`, `b`, and `c` occurs in it as a subsequence:

```
abcabac
-------
a   b c
a c b
 b a  c
 bca
  cab
  c ba
```

Your task is to find such a string for the entire English alphabet. That is, a string such that each of the 26! permutations of `a` through `z` occurs in it as a subsequence.

Input specification

There is no input.

Output specification

Your output must contain a single line with a string of lowercase English letters.
For the easy subproblem R1 we will accept any valid string that has at most 660 letters.
For the hard subproblem R2 we will accept any valid string that has at most 640 letters.
(There are valid answers shorter than 630 letters.)

