You are participating in a TV competition where your goal is to build a sequence whose last element is a given natural number . You can reach it sticking to the following rules:

1. .
2. for each , and is any of the following operations, each of which has a unique code:

|  |  |
| --- | --- |
| **Code** | **Operation** |
|  | , addition |
|  | , subtraction |
|  | , multiplication |
|  | , integer division |
|  | , remainder of division |
|  | , bitwise "and" |
|  | , bitwise "or" |
|  | , bitwise exclusive "or" |

**The constraint must be fulfilled during the whole time.**

The application of an operation is denoted by . The goal is to find a sequence of operations for which and is as small as possible.

Answer such queries.

**Input**

The first line of the file **competition.in** contains the number. Each of the next lines contains one number - for the corresponding query.

**Output**

For each query, print in the file **competition.out** first the number of operations (), and on the next lines - the operations themselves. must be satisfied. **If for some you cannot find the required operations, print .**

**Constraints**

**It is guaranteed that for every there is a solution with at most 150 operations.**

**Time limit: 5 sec.**

**Memory limit: 256 MB.**

**Scoring**

For each solved query will be added to your score, and for each . For each test, let *minScore* be the smallest score among all participants' scores and *yourScore* be your score. You will be awarded multiplied by the amount of points for the test.

The tests are distributed as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tests generation**

The numbers are randomly generated in the respective intervals that bound them (each number in the interval has an equal chance). It is guaranteed that they are distinct.

**Sample test**

|  |  |
| --- | --- |
| **Input (competition.in)** | **Output (competition.out)** |
| 2  32  123456789012345678 | 10  1 0 0  1 1 0  1 2 0  8 2 3  3 4 4  4 5 1  7 6 5  5 7 6  2 5 8  6 9 5  -1 |

**Example explanation**

For *,*

No solution was found for and is printed instead.

The total result is