BINARY SUBSEQUENCES

The scientific committee is in deep trouble: they could come up with a nice task, but they are not sure they can build the test data for it. The problem statement sounds like this:

"Being given a string containing only 0s and 1s, compute the number of distinct non-empty subsequences that appear in it"

Here, by subsequence of a string, we understand a string that can be obtained from the initial string by erasing some of its characters (possibly none). Two subsequences are considered different if and only if they differ in at least one place or they have different lengths. For example, string "101" has 6 different non-empty subsequences: "0", "1", "10", "01", "101" and "11".

Now, the committee could, of course, generate the tests randomly and compute the answer for each of them with the official source, but this idea doesn't satisfy the author. They want to have full control over the output. They want to know, for each value K in a given file, how many binary strings have **exactly** K different non-empty subsequences and what is the shortest such string. If there are more strings meeting the requirements, the committee will be pleased with any such string. As the answer to the first request might be very big, they want to know just the number modulo 1.000.000.007.

INPUT

On the first line of the input file, there will be a number T representing the number of values K that follow. On the next T lines there'll be the values of K for which you need to answer to the 2 queries described in the statement.

OUTPUT

For each value K in the input file, you'll have to print 2 lines:

On the first line, you should print the number of binary strings that have **exactly** K different non-empty subsequences modulo 1.000.000.007, or, if you want to skip this query -1. If you print any number different from -1, we'll consider that you've tried to answer the question.

On the second line, you should print -1 if you want to skip the query. Otherwise, you should print L binary digits separated by spaces, digits that should form one of the binary strings of minimal length that have **exactly** K different non-empty subsequences.

SCORE

There will be 3 subtasks for this problem. You can skip queries **only** in the first subtask by printing -1s as explained in the Output section.

Subtask	Constraints	Scoring
Subtask 1	T = 2000 and K takes all the values from 1 to 2000 in order	In order to get a non-zero score, each of your answers must be either correct, either skipped. Let A be the number of correct answers to the first requirement and B the number of correct answers to the second one. You'll get: $43 \times \frac{0.3 \times A + 0.7 \times B}{2000}$ points
Subtask 2	T <= 10, K <= 100.000	39 points
Subtask 3	T <= 10, K <= 1.000.000	18 points

EXAMPLE

input	output
3 2 3 8	2 0 0 4 1 0 -1 1 1 0 0

For K = 2, we have exactly 2 strings that with 2 different non-empty subsequences: "00" (with subsequences "0" and "00") and "11" (with subsequences "1" and "11"). To be noted that string "11" has the minimum length too and it would be accepted by the grader as a correct answer.

For K = 3, we have 4 strings fulfilling the requirement: "10" (with subsequences "0", "1" and "10"), "01" (with subsequences "0", "1" and "01"), "000" (with subsequences "0", "00" and "000"), "111" (with subsequences "1", "11" and "111")

We skipped the first query for K = 8 and "1100" has exactly 8 different non-empty subsequences: "0", "00", "1" "11", "110", "1100", "10" and "100".

This is just an example to clarify the information in the statement. As you can see, T is not 2000, so this couldn't be a test from subtask 1, which means that by skipping a query, we get 0 points.