

A. Zebras

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Oleg writes down the history of the days he lived. For each day he decides if it was good or bad. Oleg calls a non-empty sequence of days a *zebra*, if it starts with a bad day, ends with a bad day, and good and bad days are alternating in it. Let us denote bad days as 0 and good days as 1. Then, for example, sequences of days 0, 010, 01010 are zebras, while sequences 1, 0110, 0101 are not.

Oleg tells you the story of days he lived in chronological order in form of string consisting of 0 and 1. Now you are interested if it is possible to divide Oleg's life history into several **subsequences**, each of which is a zebra, and the way it can be done. Each day must belong to exactly one of the subsequences. For each of the subsequences, days forming it must be ordered chronologically. Note that subsequence does not have to be a group of consecutive days.

Input

In the only line of input data there is a non-empty string s consisting of characters 0 and 1, which describes the history of Oleg's life. Its length (denoted as $|s|$) does not exceed 200 000 characters.

Output

If there is a way to divide history into zebra subsequences, in the first line of output you should print an integer k ($1 \leq k \leq |s|$), the resulting number of subsequences. In the i -th of following k lines first print the integer l_i ($1 \leq l_i \leq |s|$), which is the length of the i -th subsequence, and then l_i indices of days forming the subsequence. Indices must follow in ascending order. Days are numbered starting from 1. Each index from 1 to n must belong to exactly one subsequence. If there is no way to divide day history into zebra subsequences, print -1 .

Subsequences may be printed in any order. If there are several solutions, you may print any of them. You do not have to minimize nor maximize the value of k .

Examples

input
0010100
output
3 3 1 3 4 3 2 5 6 1 7

input
111
output
-1