

Problem I. Addition

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 256 mebibytes

Your task is a simple one – you have to add two numbers! Unfortunately, that would be far too simple, so we had to spice things up a bit.

PAScript is a brand new script language. A script in this language consists of k lines: the i -th of them contains two non-empty words a_i, b_i , composed of printable ASCII characters (with numbers ranging from 33 to 126). The interpreter of *PAScript* reads a string s from the input and executes the following pseudocode:

```
while true:
    foundReplacement = false
    for i = 1, 2, ..., k:
        if a(i) is a substring of s:
            replace the leftmost occurrence of a(i) with b(i)
            foundReplacement = true
            break
    if not foundReplacement:
        break
print(s)
```

Your task is to find an effective script which adds two binary numbers. The input for your script will be a string of the form **binary number + binary number**, without spaces and leading zeros. An example of a correct input is 10001+1001. The two binary numbers represent non-negative integers. The length of the input string does not exceed 100.

Your script should output the sum of the input numbers in binary and without leading zeros as well. For the above example the result should be 11010.

Your script should also be short and effective. Specifically, it should satisfy the following conditions:

- $k \leq 50$ (the script can have no more than 50 lines);
- $1 \leq |a_i|, |b_i| \leq 8$ (no word in the script can have more than 8 characters nor be empty);
- For each input, the interpreter executes the external **while** loop at most 100 000 times. Moreover, during the entire execution of the script the length of s never exceeds 2017 characters.

Input

There is no input for this problem.

Output

The first line of the output should contain a single number k ($1 \leq k \leq 50$) – the number of lines in the script. The i -th of the following k lines should represent the i -th line of the script and thus should contain two non-empty words a_i, b_i separated by a single space.

Example

standard input	standard output
	6
	00 0
	01 1
	10 1
	11 1
	0 NO
	1 YES

Warning: Of course, the above example above script does not solve the task stated above. The example script, on the other hand, for a string containing zeros and ones only, finds whether it contains at least one 1.