# Task: KOD
# The Code

**XVI OI, Stage III, Day two. Source file `kod.*` Available memory: 64 MB.** *02.04.2009*

The Byteotian Institute of Telecommunication (BIT) sets standards for data transmission in telecommunication networks all over Byteotia. Byteasar, one of the engineers employed at BIT, works on *prefix codes* — a certain way of representing characters. For each and every character of the Byteotian alphabet there is a corresponding sequence of bits, called the code word of that character. The code words of all characters have the following properties:

- None of the code words is a prefix (i.e. a leading fragment) of another code word. For example, if `010010` is the code word of the letter `A`, then neither the bit sequence `0`, `01`, `010`, `0100` nor `01001` is a code word of another letter. Likewise, `0100100`, `0100101` and longer sequences starting with `010010` are not code words.
- If a given bit sequence $w$ is a prefix of another code word, but not the complete code, then each of the bit sequences $w$`0` and $w$`1` (that is, $w$ with zero or one appended at the end) is a prefix of some code word, or the complete code word. For example, if `0100` is the prefix of the code word of the letter `A`, then `01000` and `01001` each is a prefix of some code word, or a complete code word.

Let us consider the following exemplary prefix code for the alphabet consisting of the characters `A`, `B`, `C`, `D` and `E`:

| character | code word |
|---|---|
| A | 00 |
| B | 10 |
| C | 11 |
| D | 010 |
| E | 011 |

Encoding a sequence of characters with a prefix code consists in concatenating the code words of its successive characters. For example, the encoding of the sequence `BACAEBABAE` is `1000110001110001000011`.

Byteasar noticed that if some leading bits are lost, the sequence may be decoded wrong, or it may even be not decoded at all. For example, if five initial bits of the sequence given above are removed, the resulting sequence `10001110001000011` will be decoded as `BACBABAE`. The last five letter (`BABAE`) are correct, but the first three (`BAC`) are not. Byteasar further noticed that all the letters after the first `E` are decoded correct. He concluded that whenever all the bits of the code word of `E` are intact, all the characters succeeding `E` will be decoded correct. The same holds for every encoded sequence obtained from one that contains `E`. He also noticed that the letter `D` shares this feature, but `A`, `B` and `C` do not.

Because of the properties of the code words of `E` and `D` Byteasar called them *synchronising code words*. He trusted you with the task of writing a programme that finds all the synchronising code words of a given prefix code. To save some time, he intends to present you all the code words on his binary monitor. This interesting device has four buttons:

- `0` — append 0
- `1` — append 1
- `B` — backspace; deletes the last digit
- `X` — beep! the monitor beeps upon pressing this button.

Initially the display is empty; Byteasar types in successive code words by pressing the buttons, and when there is a complete code word on the screen, Byteasar presses `X`. After showing you the last code word, Byteasar clears screen by pressing `B` sufficiently many times. He declares that he will present you the complete prefix code while pressing the buttons the minimum possible number of times.

## Input

In the first line of the standard input there is an integer $n$ ($6 \leqslant n \leqslant 3\,000\,000$) denoting the number of buttons pressed by Byteasar. In the following line an $n$ letters long string consisting of the characters '`0`', '`1`', '`B`' and '`X`' is given; the characters correspond to the buttons, of course. Each time the button `X` is pressed, a code word is completed and another one starts. The code words are numbered starting with 1. The sum of lengths of all code words will not exceed $10^8$.

## Output

The number $k$ of synchronising code words should be printed out in the first line of the standard output. The following $k$ lines should contain the numbers of code words that are synchronising in increasing order, each in a separate line. If the given prefix code contains no synchronising code words, the first line should contain the number 0, and no more lines should follow.

## Example

| For the input data: | the correct result is: |
| --- | --- |
| `21` | `2` |
| `11XB0XBB00XB11XB0XBBB` | `4` |
| | `5` |

## Explanation of the example

The following code words appear successively on Byteasar's monitor: `11`, `10`, `00`, `011`, `010`. Among them, `011` and `010` are synchronising.