Code-11 is a barcode system for encoding characters used primarily in labeling telecommunications equipment. The characters encoded are limited to digits 0 through 9, the dash (``-''), and a special start/stop character which appears at the beginning and end of each Code-11 barcode.

Code-11 is a discrete system, and each character is encoded independently. A character's encoding is represented by five adjacent regions, or bars, that alternate between a dark color and a light color, starting with a dark bar. The width of each bar is either narrow or wide, according to the encoding scheme shown below, where 0 represents a narrow bar and 1 represents a wide bar.

Thus the character 1 is encoded as a wide dark bar, a narrow light bar, a narrow dark bar, a narrow light bar, and finally a wide dark bar. The barcodes for the individual characters must be separated by a narrow light bar whose only function is to separate the characters.

A two-width encoding like that used for Code-11 has the benefit of simplicity. Since it is necessary only to distinguish a narrow bar from a wide bar, Code-11 allows for a large level of print tolerance in lower-quality printing conditions.

To enable detection of errors, the Code-11 barcodes we use will have two check characters, *C* and *K*, added at the end of the message (before the stop character). If the *n* characters to be encoded (left to right) are $c_1$ through $c_n$, then the weight of the C check character is

$$\left( \sum_{i=1}^{n} ((n - i) \bmod 10 + 1) \times w(c_i) \right) \bmod 11$$

where $w(c_i)$ is the weight associated with character $c_i$. The weights for the digits 0 through 9 are 0 through 9; the weight for the hyphen is 10. (Note that mod has higher precedence than +.)

The weight of the K check character is

$$\left( \sum_{i=1}^{n+1} ((n - i + 1) \bmod 9 + 1) \times w(c_i) \right) \bmod 11$$

where $c_{n+1}$ is the C check character. For example, suppose the message to be encoded is 123-45. Then the C check character is 5 and the K check character is 2. The barcodes for the eight characters 123-4552, preceded and followed by the barcode for the start/stop character, comprise the complete Code-11 barcode encoding of the message.

Simple barcode readers measure the intensity of light reflected from a barcode to a linear array containing hundreds of tiny CCD sensors, each reporting light or dark. Light and dark regions are identified ,and the width of each region is used by the decoding software to validate the barcode and to obtain the encoded information. Since the orientation of the barcode is not fixed, the software must be able to decode the barcode whether it is scanned from left to right or from right to left.

Your problem is to decode the information obtained by scanning a Code-11 barcode, given the widths of the light and dark regions detected by the reader. Assume a wide bar is intended to be twice as wide as a narrow bar. Due to inconsistencies among printers, the width of a bar can be up to 5 percent larger or smaller than intended. There are no zero-length messages (with barcodes containing only start/stop, check characters, and intercharacter spacing).

## Input

The input contains several test cases, each representing a single scanning attempt. The data for each case begins with an integer $m \leq 150$ that specifies the number of regions detected by a barcode reader. This is followed by $m$ integers $d_1...d_m$ ($1 \leq d_i \leq 200$) giving the number of sensors in each region (within a region, all sensors report the same light intensity). The data for each test case begins and ends with a dark bar (there is no leading or trailing white space).

The last test case is followed by a single integer zero.

## Output

For each input case, display a line containing the case number and the results of the decoding effort. If the barcode can be successfully decoded, then display the sequence of message characters (without its check characters). If the decoding is successful but the C check character is incorrect, then display ``bad C''. If the decoding is successful and the C check character is correct but the K check character is incorrect, then display ``bad K''. Display ``bad code'' if the barcode cannot be successfully decoded due to bar widths outside the allowable range, missing or invalid start/stop codes, or some other invalid condition. Follow the format of the sample output.

## Sample Input

```
59
10 20 20 10 10 10 20 10 10 20
10 10 10 10 20 10 20 10 10 10
20 10 20 10 20 10 20 10 10 10
10 10 20 10 10 10 10 10 10 20
20 10 20 10 10 20 10 10 20 10
10 10 20 10 10 20 20 10 10

35
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10

35
10 10 20 20 10 10 20 10 10 10
20 10 10 20 10 10 20 10 10 10
20 10 20 10 20 10 10 10 10 10
```

```
10 10 20 20 10

0
```

## Sample Output

```
Case 1: 123-45
Case 2: bad code
Case 3: bad K
```

Harbin 2009-2010