Nekops-Nu Sequences

Nekops-Nu Sequences

A Nekops-Nu sequence is computed from any sequence of integers by following these rules starting from the first number (left-most number) in the sequence:

- Count the number of *successive* repetitions of the same number.
- Output the count and then the number.
- Repeat steps 1-3 for the next digit in the sequence

For example, given the sequence “1 2 1 1”, the Nekops-Nu sequence can be computed as:

- There is 1 instance of the number 1.
- There is 1 instance of the number 2.
- There are 2 instances of the number 1.
- The new Nekops-Nu sequence is “1 1 1 2 2 1”

The same set of rules can be applied to the resulting sequence to generate more patterns. See the sample output for further continuations to the example provided.

Task

Write a program that computes the next K instances of the Nekops-Nu sequence for a given sequence of numbers.

Input

The input will contain:

- 0 <= Number of iterative sequences to compute, K <= 10
- Initial sequence of 1 <= N <= 250
- The numbers in the sequence can have any valid 32-bit integer value.

Output

The output contains one Nekops-Nu sequence per line starting with the initial input sequence and followed by the K sequences computed. The numbers in each sequence are separated by a single space character. To make the output look more interesting, add periods to pad all the sequences (before and after) such that they are centered align with respect to the longest sequence computed. Given the example above, the two sequences were “1 2 1 1” (length 7 characters) and “1 1 1 2 2 1” (length 11 characters). Therefore the output should show:
..1 2 1 1..

1 1 1 2 2 1

If a specific row and the longest row do not line up (because of odd and even numbers of characters), then add an extra dot **before the sequence** as shown in input/output #2, or refer to the following example:

.....9999 9999....
.....2 9999.....
....1 2 1 9999....
1 1 1 2 1 1 1 9999

(i.e. if the number of characters is even in a specific row, then there should be an extra dot before the sequence)

Finally, on the last line, output the number of elements found in the last sequence.

Note: There is a newline character at the end of the last line of the output.

Sample Input 1

---

3 1 2 1 1

Sample Output 1

....1 2 1 1....
..1 1 1 2 2 1..
..3 1 2 2 1 1..
1 3 1 1 2 2 2 1
8

Sample Input 2

---

1 1 1 1 1 1 1 1 1 1 1

Sample Output 2

---

1 1 1 1 1 1 1 1 1 1
.....10 1.....
2