CODE LIKE HUFFMAN

CHALLENGE DESCRIPTION:

One of the first algorithms for efficient information coding was proposed by David A. Huffman in 1952. The idea of the algorithm is the following: knowing the estimated probability or frequency of occurrence for each possible value of the source symbol, we can describe the procedure for building variable-length code table for encoding a source symbol.
This algorithm is currently used in many data compression applications, including this challenge.

INPUT SAMPLE:

The first argument is a path to a file. Each line includes a test case which contains letters from which you have to build a code table for each character using Huffman algorithm.

```
abc
ilovecodeeval
```

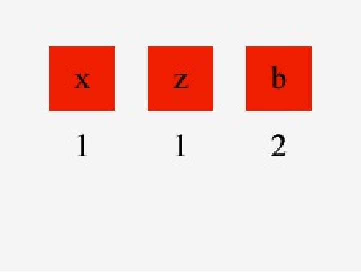
OUTPUT SAMPLE:

Use this algorithm to build a code table for each character and print it in an alphabetical order in the following way.

```
a: 10; b: 11; c: 0;
a: 1000; c: 1001; d: 1010; e: 01; i: 1011; l: 110; o: 111; v: 00;
```

CONSTRAINTS:

1. The test case can include lowercase characters only.
2. When building a binary tree, if the priority of items is the same, the sorting should be done in an alphabetical order, that is:

   If the priority of items is the same then Node has higher priority than symbol. If 2 Nodes have same priority then sorting should be done in an alphabetical order.
3. Test case can include from 3 to 30 characters.
4. The number of test cases is 40.