Problem B

Computer filesystems are often filled with multiple copies of identical files. These identical files take up unnecessary space, since one copy, plus appropriate file system links, are sufficient to represent all the copies. A simple way to combat this problem is to compare all pairs of files, removing all duplicates.

Comparing all pairs of nn files requires n(n−1)/2n(n−1)/2 comparisons, which can be quite large, since nn may be very large. Compounding the problem is the fact that comparing two files is slow, especially if they are large and mostly the same.

One way to reduce this cost is to convert each file to a short string (called a hash) using a deterministic hash function, and then compare these hashes (instead of whole files) to identify potential matches. If two hashes don’t match, neither do their corresponding files. If two hashes do match, it may be because their corresponding files are identical, or because there is a hash collision (two different files that produce the same hash). The most straightforward way to determine this is to compare the files directly.

For this problem, write a program which determines file duplicates using hashing (to identify potential duplicates and eliminate impossible ones). The hash function is simple, taking as input the entire file, and producing as output one byte. The output is the exclusive or (XOR) of the ASCII value of every byte in the input.

**Input**

Input consists of up to 250250 test cases. Each test case begins with an integer 1≤n≤5001≤n≤500 indicating the number of files. This is followed by nn lines, each representing one file. Each line has 11 to 5050 characters, using only the characters a–z, A–Z, space, and period (.). Input ends when n=0n=0.

**Output**

For each test case, print the number of unique files and the number of hash collisions between all pairs of files.

|  |  |
| --- | --- |
| **Sample Input 1** | **Sample Output 1** |
| 4  four score and seven years ago  score four and seven years ago  four score and seven years ago  ask not what your country can do for you  4  the quick brown fox jumped over the lazy dog  over the lazy dog the quick brown fox jumped  the lazy quick fox jumped over the brown dog  the quick lazy dog over the brown fox jumped  0 | 3 2  4 6 |