4929 Finite Fingerprints

As a scientist, you regularly deal with data collected from many ongoing experiments. The data each experiment produces is a very long binary sequence, where each bit represents a result (0 or 1) at a particular time. You have a suspicion that many of the experiments demonstrate cyclic behavior with a particular period. That is, the data they produce repeats at regular time intervals. However, you're not certain. You want to analyze the data and see if they support your hunch.

One of your staff has taken multiple samples (called 'finite fingerprints') from each experiment's data sequence. Each fingerprint is of the same length - the length of the suspected period. It is convenient that each fingerprint's length is always a multiple of six, because it allows your staff member to encode the fingerprint in base-64 to compress it and send it to you.

You know that if an experiment produces cyclic data, then any two fingerprints from its sequence will have exactly the same order of bits. However, they might be time-shifted because the fingerprints started at different times during the cycle. For example, suppose that the underlying sequence is cyclic and looks like this:

$$\ldots 001011001011001011001011 \ldots$$

Then if we take 3 fingerprints at random from this sequence of length 6, we might get 001011, 011001, and 100101. All of these can be cyclically shifted (i.e. move bits from the end to the beginning, or vice-versa) so that they are identical. Thus, these fingerprints support your theory.

Write a program that will help you analyze which fingerprints might represent the same cyclic behavior and group them together.

Base-64 encodes six bits in each character, using big-endian encoding (the most significant bit is given first). The characters representing these values are (in order) `A-Z`, `a-z`, `0-9`, `+`, and `/`. Thus, the string '`At4/`' represents the bit sequence '000000 101101 111000 111111' (with spaces added for clarity).

Input

Each test case will begin with an integer $0 < n \leq 1000$, indicating the number of fingerprints. This is followed by n fingerprints with corresponding indexes 1 through n. Each fingerprint is given in base-64 encoding, and all N have the same length. No fingerprint will be longer than 1000 base-64 characters. End of input is signaled by $n = 0$.

Output

For each test case, print the case number. Then for each group of fingerprints, print the members of each group using their input indexes (1 through n), separated by spaces. Each case will be on a separate line. Order the groups by the number of members (smallest to largest), breaking ties by the group with the lowest-indexed fingerprint. Within each group, order the members by index.

Sample Input

```
7
AAEA
Bbx+
C3j8
AAAA
FvH4
```

```
At4/
AAAB
10
IfgTLhobrrOylBGMaYMH
w0N11nZTAjGNMGDkPwJl
lBGMaYMHIfgTLhobrrOy
EYxpgwch+BMuGhuus7KY
AmXDQ3XWdlMCMYOwYOQ/
oIxjTBg5D8CZcNDddZ2U
4Ey4aG66zspQRjGmDByH
BGMaYMHIfgTLhobrrOyl
ch+BMuGhuus7KYEYxpgw
Ghuus7KYEYxpgwch+BMu
0
```

Sample Output

```
Case 1:
4
1 7
2 3 5 6
Case 2:
1 3 6 7 8
2 4 5 9 10
```