

`strcmp()` is a library function in C/C++ which compares two strings. It takes two strings as input parameter and decides which one is lexicographically larger or smaller: If the first string is greater then it returns a positive value, if the second string is greater it returns a negative value and if two strings are equal it returns a zero. The code that is used to compare two strings in C/C++ library is shown below:

```
int strcmp(char *s, char *t)
{
    int i;
    for (i=0; s[i]==t[i]; i++)
        if (s[i]=='\0')
            return 0;
    return s[i] - t[i];
}
```

Figure: The standard `strcmp()` code provided for this problem.

The number of comparisons required to compare two strings in `strcmp()` function is never returned by the function. But for this problem you will have to do just that at a larger scale. `strcmp()` function continues to compare characters in the same position of the two strings until two different characters are found or both strings come to an end. Of course it assumes that last character of a string is a null (`'\0'`) character. For example the table below shows what happens when “than” and “that”; “therE” and “the” are compared using `strcmp()` function. To understand how 7 comparisons are needed in both cases please consult the code block given above.

t h a N \0 = = = ≠ t h a T \0 Returns negative value 7 Comparisons	t h e r E \0 = = = ≠ t h e \0 Returns positive value 7 Comparisons
--	--

Input

The input file contains maximum 10 sets of inputs. The description of each set is given below:

Each set starts with an integer N ($0 < N < 4001$) which denotes the total number of strings. Each of the next N lines contains one string. Strings contain only alphanumerals (`'0'... '9'`, `'A'... 'Z'`, `'a'... 'z'`) have a maximum length of 1000, and a minimum length of 1.

Input is terminated by a line containing a single zero.

Output

For each set of input produce one line of output. This line contains the serial of output followed by an integer T . This T denotes the total number of comparisons that are required in the `strcmp()` function if all the strings are compared with one another exactly once. So for N strings the function `strcmp()` will be called exactly $\frac{N(N-1)}{2}$ times. You have to calculate total number of comparisons inside the `strcmp()` function in those $\frac{N(N-1)}{2}$ calls. You can assume that the value of T will fit safely in a 64-bit signed integer. Please note that the most straightforward solution (Worst Case Complexity $O(N^2 * 1000)$) will time out for this problem.

Sample Input

```
2
a
b
4
cat
hat
mat
sir
0
```

Sample Output

```
Case 1: 1
Case 2: 6
```