Since you crave state-of-the-art technology, you've just purchased a phone with a great new feature: autocomplete! Your phone's version of autocomplete has some pros and cons. On the one hand, it's very cautious. It only autocompletes a word when it knows exactly what you're trying to write. On the other hand, you have to teach it every word you want to use.

You have N distinct words that you'd like to send in a text message in order. Before sending each word, you add it to your phone's dictionary. Then, you write the smallest non-empty prefix of the word necessary for your phone to autocomplete the word. This prefix must either be the whole word, or a prefix which is not a prefix of any other word yet in the dictionary.

What's the minimum number of letters you must type to send all N words?

Input: Input begins with an integer T, the number of test cases. For each test case, there is first a line containing the integer N. Then, N lines follow, each containing a word to send in the order you wish to send them.

Output: For the ith test case, print a line containing "Case #i: " followed by the minimum number of characters you need to type in your text message.

Constraints: 1 ≤ T ≤ 100, 1 ≤ N ≤ 100,000, The N words will have a total length of no more than 1,000,000 characters. The words are made up of only lower-case alphabetic characters. The words are pairwise distinct.

NOTE: The input file is about 10-20MB.

Sample:

Input:

```
1
5
hi
hello
lol
hills
hill
```

Output:

```
Case #1: 11
```

You will write "h", "he", "l", "hil", "hill", for a total of 1 + 2 + 1 + 3 + 4 = 11 characters.

4682 - XOR Sum

Asia - Amritapuri - 2009/2010

Given an array of N numbers, we wish to choose a contiguous sub-sequence of the array, so that the bitwise XOR of all chosen numbers is maximum. Bitwise XOR is defined as follows: every bit in the answer is obtained by applying XOR logic on the corresponding bits of the set of numbers. For example 7, 8 and 5 are XORed as follows,

```
Numbers in binary:  0111  1000  0101  -----  1010
```

So the answer is 10 (in decimal). The same answer can be obtained in C/C++/Java by using the XOR operator as 7^8^5.

Input

The first line contains the number of test cases T. The first line of each test-case contains one integer, **N** (size of the array). The next **N** lines of each test-case contain integers denoting the elements of the array.

Constraints:

- $1 \leq T \leq 10$
- $1 \leq N \leq 100,000$
- All input integers will be non-negative and fit into 32 bit signed integer.

Output

For each test case, output a single line containing the maximum sum that can be obtained.

Sample Input

```
2  5  3  7  7  7  0  5  3  8  2  6  4
```

Sample Output

```
7  15
```

Amritapuri 2009-2010

Problemsetter: Prasanna Sankaranarayanan
Special Thanks: Adrian Kuegel, Harsha Suryanarayana