
Judicious Strategy

Input file: `stdin`
Output file: `stdout`
Time limit: 1 seconds
Memory limit: 128 megabytes

Alice and Bob is now playing a game about strings.

There is a dictionary containing n words (words might be same). Alice choose a lowercase English letter arbitrarily first, but this letter should appear in at least one of these n words. Then Bob choose a lowercase English letter arbitrarily to add it before or after the letter Alice chose. So Bob gets a new string now. This new string should also be a substring (consecutive subsequence) of at least one strings in the dictionary. After that, it's Alice's turn. Alice should do the same thing, choosing a letter and add it before or after the current string, making a new string. At every moment, the string they made should always be a substring of at least one strings in the dictionary. The player who can't operate first lose the game and the other one win.

Besides, each player has a score. The score is calculated by the following rule:

If the string S is now made, the current player will get $score(S)$ points. It means that Alice will score in the first round, then Bob, then Alice...

$$score(S) = \left[\left(\sum_{i=1}^{|S|} value(S_i) \right) \times \max_{i=1}^{|S|} value(S_i) \right] + occ(S)$$

where

- $|S|$ means the length of S .
- $value(c)$ represents the value of letter c . The score of letter "a" is 1, "b" is 2, ..., "z" is 26.
- $occ(S)$ means the time that S occurs as a substring in the dictionary, each word is counted just once.

Alice and Bob will play with best strategy. That is to say, they will consider to win first and then maximize their score, after that they will consider to minimize the score of others.

Please determine who will win the game, and report the final scores they will earn during the whole game.

Input

The input contains several test cases, no more than 10 test cases.

In each test case, the first line contains an integer $n(1 \leq n \leq 30)$, denoting the number of words in the dictionary.

In the next n lines, each line contains a non-empty string $word_i$, denoting a word in the dictionary. The string is composed of lowercase English letters and its length will not exceed 30.

Output

For each test case, output a string in the first line. If Alice will win, output “Alice”, otherwise output “Bob”.

Then print two integers A and B in second line, denoting the final score of Alice and Bob.

Examples

stdin	stdout
2 aba abac	Bob 29 35
3 artem nik max	Alice 2403 1882