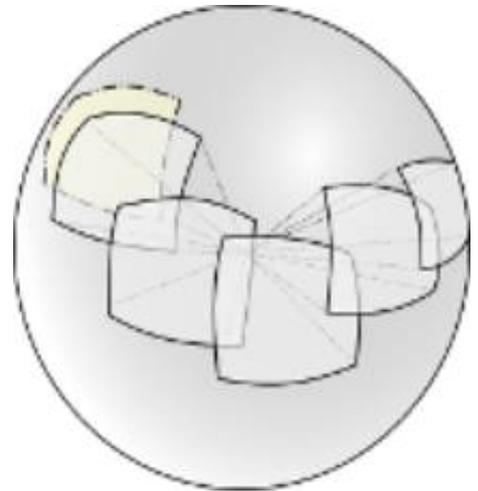


## 4994 Overlapping Scenes

The Dummywood film industry is infamous for its repetitive stories. The phrase, “if you have seen one, you have seen them all” perfectly applies to it. The movies are made by fitting together existing set of scenes. This reduces production time, as there is hardly ever a new stunt to perform, nor the writers need to be creative in bringing out something innovative. To make the movies lengthy, the producers opt to repeat monotonous events, such as repetitive breaking of glass doors or the “bad guys” rolling over the floor. Cutpiece is a producer who wants to use the power of computers to gain advantage in this film industry. He has a set of scenes out of which he wants to make his next movie. He plans to merge these scenes to make one complete movie. Although, mere merging of the scenes in some order would make a movie, but Cutpiece wants to minimize the length of the movie. The minimizing technique that he plans to apply relies on the fact that, scenes are so repetitive in their components that, the beginning of one and ending of another may be identical up to a certain length. Cutpiece has decided to condense these scenes so that the repetitive portions are included once in the merging process. In this problem, given a set of scenes, you will have to determine the minimum length of the movie that can be made by merging the given scenes in a particular order, condensing the repetitive portions during the merging process. Look at the explanation for sample input/output below to further clarify the condensing process.



### Input

The first line of input will consist of a positive integer  $T \leq 50$ , where  $T$  denotes the number of test cases. Each case starts with a positive integer  $n \leq 6$  where  $n$  denotes the number of scenes that Cutpiece will merge. The following  $n$  lines will each contain a string of length at most 10. The strings will consist of upper case letters only and will have at least one character. Each of these strings represents one scene and the individual letters correspond to components forming a scene.

### Output

For each case of input, there will be one line of output. It will consist of the case number followed by the minimum length of movie that can be made. Look at the output for sample input for exact formatting.

### Explanation of Sample Input/Output:

Case 1 -> if we order the input strings as "ABCD" "CDEF" and "DEFGH". Merge the first 2 to get "ABCDEF". Here we condense (CD) into a single occurrence since this is the longest length common suffix of one and prefix of another. Next we merge ABCDEF with DEFGH to obtain ABCDEFGH, giving us a string of length 8. Any other ordering of the three strings will not yield a shorter final string. Note that, when merging, we always merge from left to right after ordering the string. Therefore, for the above ordering, we would not merge CDEF and DEFGH first.

Case 2 -> Here one string is a subset of another and we can entirely condense the components of shorter string to give us a string of length 7.

**Sample Input**

```
2
3
ABCD
DEFGH
CDEF
2
AAAAA
AAAAAAA
```

**Sample Output**

```
Case 1: 8
Case 2: 7
```