When we're not fighting Internet bad guys at Sift, we can often be found playing board games. The goal of this challenge is to write a program to help us play the game of SET (http://www.setgame.com/). In particular, we want to find the largest number of disjoint SETs in a collection of SET cards.

Input

SET is played with a special deck of cards. Each card has four attributes:

1. Color
2. Symbol
3. Shading (of symbols)
4. Number (of symbols)

Each attribute has three potential values and each card is unique, so there are a total of 81 (3^4) possible cards.

In the input to our program, the four attributes of a SET card will be represented as follows:

1. Color -- one of the words "blue", "green", or "yellow"
2. Symbol -- the three symbols will be represented by the letters A, S, and H
3. Shading -- the shading of the symbols will be represented by the case of the letter: one of lower-case (a,s,h), upper-case (A, S, H), or "symbol-case" (@, $, #)
4. Number -- the letter will appear one, two, or three times

For example, possible SET cards include:

- **`yellow S`** -- one Yellow upper-case S
- **`blue aaa`** -- three Blue lower-case A's
- **`green ##`** -- two Green symbol-case H's

The input to our program will be through standard input (not through a file) and will contain an integer N, with 3 <= N <= 30, followed by a list of N distinct SET cards, one per line. For example:

```
15
blue #
green $
blue AA
yellow @
blue @@@
green A
yellow $$$
yellow @@@
yellow HHH
yellow #
yellow @@
blue a
blue sss
green a
green @
```

What is a SET?

A set of three cards forms a SET if (and only if), for each of the four attributes, the three cards either all have the same attribute value or all have different values. For example, the three cards **`green $, yellow SS, and blue sss`** form a set, because:

1. **t**he three cards have different colors (green, yellow, and blue)
2. the three cards have the same symbol "S"
3. the three cards have different cases (symbol-case, upper-case, and lower-case)
4. the three cards have different numbers of symbols (one, two, and three)

Additional example of SETs:

- **`blue H, green S, yellow A`** -- the cards all have the same number (1) and case (upper-case), while all having different colors and symbols.
- **`green $, green $$, green $$$`** -- the cards have the same color (green), symbol (S), and case (symbol-case), while all having a different number of symbols.

On the other hand, none of the following are SETs:

- **`yellow $, blue H, green aa`** -- the first and second cards have one symbol each, while the third card has two symbols.
- **`green HHH, blue hhh, blue HHH`** -- the second and third cards are both blue, while the first card is green. Further, the first and third cards are both have upper-case symbols, while the second card is lower-case.

An alternate way of identifying SETs may be helpful: three cards do **not** form a SET if (and only if), for any of the four attributes, two cards have the same value but the third is different.

Goal

The two goals for this challenge are to read a collection of SET cards from standard input (stdin) and then to:

1. find and print the number of possible SETs of three cards in the input, and
2. find and print the largest disjoint collection of SETs in the input
   (any correct answer can be output if there are multiple, maximum-size collections).

Two SETs are "disjoint" when they contain no common cards. For example, the two SETs:

`green $, yellow SS, blue sss`, and
`blue H, green S, yellow A`

are disjoint, while SETs:

`green $, yellow SS, blue sss`, and
`green $, green $$, green $$$`

are ***not*** disjoint, as both contain the card **`green $`**.

In our example input, there are a total of 9 sets, and the maximum number of disjoint sets is 4.

Output

The program's output should consist of:

1. A single line containing the number of possible SETs of three cards in the input.
2. A single line containing the maximum number of disjoint SETs in the input.
3. The cards forming a largest collection of disjoint SETs, each card on its own line and each SET preceded by a blank line.

For our example input above, an acceptable output would be:

```
9
4

blue #
green $
yellow @

blue a
blue AA
blue @@@

green a
green A
green @

yellow #
yellow @@
yellow $$$
```

Frequently Asked Questions

1. *How long can my code run on an input?*
   Your solution should run for no longer than 5 seconds on a test input.

2. *How should I read the input?*
   Your program should read the input values through standard input, for example `cat input.txt | [command to run your solution]`. Do not hardcode or prompt for a file name.

3. *Can I assume that the test input will be valid?*
   Yes.

4. *What programming language should I use?*
   You can use any language you feel comfortable with.

5. *Can I use third party libraries?*
   Please use only your language's standard libraries for the core logic of your solution.

6. *Should I include any test files in my submission?*
   Test files are not required, but please include any tests or test code you wrote and clear instructions on how to run them.

7. *When do I have to submit the take home test?*
   You will have 72 hours to complete the take home test and submit it.

8. *How should I submit my solution?*
   You should submit a zipped file via Greenhouse containing your solution, a README file with instructions on how to run your solution code, and any test file you might have to validate your solution.

9. *Do you have any larger example inputs?*

Example Input

```
28
blue hhh
yellow @
green ##
yellow ###
blue AA
green SSS
blue ###
yellow s
yellow ##
blue H
green A
blue $
green SS
green ###
blue ss
yellow $
green aaa
green AA
yellow sss
green aa
green S
green HH
yellow AA
yellow ss
green h
```

Example Output

```
41
8

green aaa
blue hhh
yellow sss

blue H
green SSS
yellow AA

green AA
yellow s
blue ###

green h
green aa
green sss

blue ss
green A
yellow ###

blue $$
green ###
yellow @
```

blue $$
blue aa
green sss

blue AA
yellow ss
green ##

blue aa
yellow ##
green SS