Problem 4 – Jedi Dreams

Now, here is Ivan. He is a young padawan. He thinks that, he can learn how to use the Force, but that is not true. He dreams about how one day, he will hack The Dark Side Judge System and become a Master Jedi.

Your task is simple. Help Ivan to understand how the Force works and hack the Judge together.

Actually, you need to solve one hard programming problem.

You are given a source code with valid syntax. Your task is to find where all method invokes happen. For each declared method, you need to find all invoked methods in it. See the example below for better clarification.

For your convenience the code, has some limitations to make the task easier.

- The code will be fully understandable by your current level at Java Jedi Part 2. There will be no strange structures, object oriented programming, unknown keywords, whatsoever…
- **All method declarations will be static** without any access modifiers such as "public", "private" and "protected".
- The code will not be **necessary compiling** but will be with valid syntax.
- **All method names** will be on the same line with the **static** keyword.
- There will **not** be any other static declarations **except for the methods**.
- There will **not** be any **commented code or code in strings**.
- Brackets are your best friends.

You will be given **N** lines with source code. Find **all** method invokes in a particular method declaration.

Each declared method must be **ordered** by the **count** of the methods invoked in it – **descending order**. If two declared methods are with the same count of inner calls, print them in **alphabetical order**. All invoked methods must be sorted **alphabetically**. Print them in the following format:

"{method name} -> {number of calls} -> {invoked method 1, invoked method2, …}"

If there are no invoked methods in certain declaration, print:

"{method name} -> None"

See the examples below for more information.

Input

- On the first input line you will be given the number **N**.
- On the next **N** lines you will be given lines of code.

Output

- The output data should be printed on the console.
- Each method declaration must be printed on a separate line, in which you show all invoked methods in it in the format described above.

Constraints

- **N** will be between 10 and 200, inclusive.
- The input data will **always be valid** and in the format described. There is no need to check it explicitly.
- All method names will contains at least one **uppercase** character.

Example Input

```
46
static void InitPatterns(List<bool[,]> patterns)
        {
            // zero
            patterns.Add(new bool[,]
            {
                //TODO
            });

            // one
            patterns.Add(new bool[,]
            {
                //TODO
            });
        }

        static bool CheckCurrentPattern(int[,] numbers , bool[,] pattern , int row , int col , int digit)
        {
            for(int i = 0; i < pattern.GetLength(0); i++)
            {
                for(int j = 0; j < pattern.GetLength(1); j++)
                {
                    //TODO
                }
            }

            return true;
        }

        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());

            int[,] numbers = new int[n , n];

            for(int i = 0; i < n; i++)
            {
                var currentNumbers = Console.ReadLine().Split(' ')

                for(int j = 0; j < currentNumbers.Length; j++)
                {
                    numbers[i , j] = int.Parse(currentNumbers[j]);
                }
            }

            List<bool[,]> patterns = new List<bool[,]>();
            InitPatterns(patterns);
```

Example Output

```
Main -> 6 -> InitPatterns, Parse, Parse, ReadLine, ReadLine, Split
CheckCurrentPattern -> 2 -> GetLength, GetLength
InitPatterns -> 2 -> Add, Add
```