Matermind Game

Mastermind is a puzzle game in which one player (the *codebreaker)* tries to guess the code their opponent (the *codemaker*) comes up with. The game is played on a board with two kinds of pegs:

1. Big colorful pegs used by both players:
   to form the code, as well as for the guesses.

2. Small black and white pegs, used by the *codemaker* to give feedback to the *codebreaker.*

The *codemaker* chooses a pattern of four code pegs (duplicates are allowed). The chosen pattern is placed in the four holes covered by the shield, visible to the *codemaker* but not to the *codebreaker*.

The *codebreaker* tries to guess the pattern, in both order and color. Each guess is made by placing a row of code pegs on the decoding board.

Once placed, the *codemaker* provides feedback by placing 0-4 key pegs in the small holes of the row with the guess. A black key peg is placed for each code peg from the guess which is correct in both color and position. A white key peg indicates the existence of a correct color code peg placed in the wrong position.

The order of these small B&W key pegs doesn't matter.

See here for the full details of the game here: https://bit.ly/2ZtfXpg.

For our exercise, we will build a function that takes a list of 4 colors, and returns a list of 0-4 black/white results.

Part of this exercise is reading comprehension and domain discovery, so we won't outline all forms of input and output here. Read the rules above (and the linked post) to familiarize yourself with the game's rules, so that your implementation is consistent with the game's rules.

The objective is to compare the actual color-code to the guessed color-code, and return the correct amount of black/white pegs (based on the game's rules).

Your implementation should be as efficient as possible, clean and easy to read & understand.

(See the attached C# file for the skeleton code)

```
class Mastermind
{
    private List<CodePeg> code;

    public Mastermind(List<CodePeg> code)
    {
        this.code = code;
    }

    public List<ResultPeg> GetHints(List<CodePeg> guess)
    {
        // Implement the logic here
    }
}
```