In this assignment, you will be writing a computer player for the game of Mastermind, using the ILOG constraint programming library. Mastermind is traditionally a board game for two players. The goal of the game is for the code-breaker, which is implemented by you, to guess the secret code chosen by the code-maker, which is provided in the support code. The code is an ordered sequence of four colored pegs chosen from six available colors. The code-breaker makes a series of guesses about the pattern. The code-maker provides feedback to each guess through two numbers, the number of pegs that are of the right color and position, and the number of pegs that are of the right color but not the right position. These are represented by black and white pegs, respectively, that the code-maker provides in response to the code-breaker's guess.

Your program should be able to solve mastermind games with a code word of size 4 and with 6 colors. If you wish to experiment with larger codewords/more colors, feel free to change the constants in `Mastermind.h` in the support code.

There are many freely available versions of Mastermind online, which are worth spending some time with if you've never played Mastermind before:

- `http://www.archimedes-lab.org/atelier.html?http://www.archimedes-lab.org/mastermind.html`
- `http://www.scugog-net.com/room108/masterm/master.html`

In our variant of mastermind, the solution may contain a repetition of colors.

Provided Environment

The support code is segmented into three pieces. First, there is the **CodeMaker** class, which creates a new secret pattern, and can be queried for responses to each guess you make. There are also the actual game definitions and structures for interaction between the **CodeMaker** and **CodeBreaker**, which are in the file `Mastermind.H`. Additionally, a `main.C` file has been provided, which will print out your successive guesses in addition to the computer's corresponding feedback.

Your Tasks

You must write the third part of the code base, the **CodeBreaker** class. The code breaker is responsible for providing a guess when queried, and is updated with the **CodeMaker**'s feedback. You should start with an initial guess, and through successive feedback, create and refine a constraint program that is solved by ILOG Solver to provide the next guess.

Support Code

When you are ready to begin coding, copy the support code from `/course/cs149/asgn/mastermind/`.
You can compile by typing:
`cslab0a /course/cs149/asgn/mastermind% make`
and run by typing:
`cslab0a /course/cs149/asgn/mastermind% ./mastermind`
Handin by typing:
`cslab0a /u/sello/course/cs149/mastermind% /course/cs149/bin/cs149_handin mastermind`

Documentation

ILOG Solver example files:
`file:///com/solver/examples/src/`
Solver reference:
`file:///com/solver/doc/`