Department of Electrical and Computer Engineering
EECE 2560: Fundamentals of Engineering Algorithms

Project #1

In this project, you will write a complete program that allows the user to play a game of Mastermind against the computer. A Mastermind game has the following steps:

1. The codemaker selects a code, a random sequence of 4 digits, each of which is in the range [0,5].

2. The codebreaker is prompted to enter a guess, a 4-digit sequence.

3. The codemaker responds by printing two values that indicate how close the guess is to the code. The first response value is the number of digits that are the right digit in the right location. The second response value is the number of digits that are the right digit in the wrong location.

   For example if the secret code is $(1, 2, 3, 4)$ and the guess is $(4, 0, 3, 2)$, the response would be $(1, 2)$ because one digit (3) is the right digit in the right location, and two digits (2 and 4) are the right digits in the wrong locations.

   Note that no digit in the code or guess is counted more than once. If the secret code is $(1, 2, 3, 4)$ and the guess is $(2, 1, 2, 2)$, the response is $(0, 2)$. If the secret code is $(3, 2, 3, 3)$ and the guess is $(1, 3, 3, 4)$, the response is $(1, 1)$.

4. The codebreaker is prompted to continue entering guesses. The codebreaker wins if the correct code is guessed in ten or fewer guesses, and otherwise the codemaker wins.

Your programs should be modular and should make full use of object-oriented programming techniques. Each class should clearly separate its interface from its implementation. Use member functions to implement all commonly used operations. Every function should be documented to describe its function, assumptions and limitations. Each function should validate its inputs and print error messages if the inputs are not correct.

Implement a version of the game in which the computer is the codemaker and a human is the codebreaker.

1. Implement the class `code` which stores the code (either a secret code or a guess) as a vector, and which includes:

   (a) the `code` class declaration,

   (b) a function that initializes the code randomly

   (c) a function `checkCorrect` which is passed a guess as a parameter, i.e. another `code` object, and which returns the number of correct digits in the correct location,

   (d) a function `checkIncorrect` which is passed a guess as a parameter (i.e. another `code object` and returns the number of correct digits in the incorrect location. *No digit in the guess or the code should be counted more than once.*

2. Implement the class `response` which stores the response to a guess (number correct and number incorrect), and which includes:

(a) constructors,

(b) functions to set and get the individual stored values

(c) a function that compares responses and returns true if they are equal

(d) a function that prints a response.

3. Implement the class `mastermind` which handles the mechanics of playing the game, and which includes:

   (a) a `code` object as a data member,

   (b) a function that prints the secret code,

   (c) function `humanGuess()` that reads a guess from the keyboard and returns a code object that represents the guess,

   (d) a function `getResponse()` that is passed two codes (a guess and the secret code), and returns a response

   (e) a function `isSolved()` that is passed a response and returns true if the response indicates that the board has been solved.

   (f) a function `playGame()` that initializes a random code, prints it to the screen, and then iteratively gets a guess from the user and prints the response until either the codemaker or the codebreaker has won.

4. Implement a function `main()` which initializes a `mastermind` object and then calls `playGame()`.