

Tutorial 1: Cows and Bulls

120.3: Programming III, C

1 Cows and Bulls

Cows and Bulls is a game played between two players, the scorer and the guesser. The scorer chooses a list of 4 numbers (repetitions are not allowed) from the numbers 1, 2, 3, 4, 5, 6, 7, 8 and 9. For example a valid list is [1, 8, 2, 4]. The scorer keeps this list secret: it is called the code. The guesser now tries to guess the code. The scorer gives a score to each guess the guesser makes. This score is given in the form (no. of black pegs, no. of white pegs). One black peg is scored for each number that is correct and in the correct place. One white peg is scored for each colour that is correct but in the wrong place.

For example if the scorer's secret code is [1, 8, 2, 4] and the guesser guesses [1, 2, 3, 4] then the scorer should award a score of (2,1). The two black pegs are for getting 1 and 4 in the right place, and the white peg is for 2 which is in position 3 in the guess and position 2 in the secret code. The 8 does not appear at all in the code and so earns nothing towards the score. The guesser tries to guess the code in as few tries as possible by making use of the information provided by the scores of past guesses. A correct guess of course earns (4,0) and the game is then over.

A simple, static Java implementation of Cows and Bulls follows. It allows a guesser to play up to 5 games against the computer. The implementation is as follows:

- The predicate `anotherGame` asks the user whether another game is wanted and returns `true` if and only if the user replies with y or Y.
- The method `readGuess` reads in 4 numbers for the guess from the keyboard and returns the guess.
- The method `blackScore` takes as arguments a code and a guess and returns the number of positions that the code and guess are identical.
- The method `whiteScore` takes as arguments a code and a guess and returns the number of matches of code to guess which are in different positions.
- The method `printScore` takes as arguments a code and a guess and prints on the screen (b,w) where b is the black score and w is the white score.

The task of this tutorial is for you to convert the Java code into C code, compile and run it. The aim is to get you started in writing C programs by using concepts you already understand. Note that some of the code can be converted directly. In other places you will need to think about how to get around the limitations in the C syntax presented so far such as:

- The inability to define the return type of `readGuess` as a char array.
- No equivalent to the `nextInt` function.

```

import java.util.Scanner;

public class CowsAndBulls {
    public static boolean anotherGame(Scanner s) {
        String answer;

        do {
            System.out.print("Do you want to play another game? [y/n]: ");
            answer = s.next().toLowerCase();
        } while (!(answer.equals("y") || answer.equals("n")));

        return answer.equals("y");
    }

    public static int[] readGuess(Scanner s) {
        System.out.print("Enter your guess: ");

        int[] guess = new int[4];
        for (int i = 0; i < 4; i++) {
            do {
                guess[i] = s.nextInt();
            } while (guess[i] < 1 || guess[i] > 9);
        }

        return guess;
    }

    public static int blackScore(int[] guess, int[] code) {
        assert (guess.length == code.length):
            "guess and code should have the same length";

        int score = 0;
        for (int i = 0; i < guess.length; i++) {
            if (code[i] == guess[i]) {
                score++;
            }
        }

        return score;
    }

    public static int whiteScore(int[] guess, int[] code) {
        assert (guess.length == code.length):
            "guess and code should have the same length";

        int score = 0;
        for (int i = 0; i < guess.length; i++) {
            for (int j = 0; j < guess.length; j++) {
                if (i != j && code[i] == guess[j]) {
                    score++;
                }
            }
        }

        return score;
    }

    public static void printScore(int[] g, int[] c) {

```

```

        System.out.println("(" + blackScore(g, c) + ", " + whiteScore(g, c)
            + ")");
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        int[][] codes =
            {{1, 8, 9, 2}, {2, 4, 6, 8}, {1, 9, 8, 3}, {7, 4, 2, 1}, {4, 6, 8, 9}};

        for (int i = 0; i < codes.length; i++) {
            int[] guess = readGuess(s);
            while (blackScore(guess, codes[i]) != codes[i].length) {
                printScore(guess, codes[i]);
                guess = readGuess(s);
            }

            System.out.println("You have guessed correctly!");
            if (i < codes.length - 1) {
                boolean another = anotherGame(s);
                if (!another) {
                    break;
                }
            }
        }
    }
}

```