Hangman:

Hangman class: Already provided for you:

data fields:
1. words - array of words to guess (each of length 7)
2. numWords - length of words array
3. gameWord - word to guess
4. dispArr - current state of guess word
   - initially: [---[---[---[---[---[---[---]

methods:
1. showMan (int numParts) - displays hangman
2. DefaultWords() - sets the array of guess words
3. Set/ get methods for all data fields.
   (may not need all of these)

You write:
In Hangman:

1. SelectGameWord(): randomly select a word from words array
2. GetCurrentWord(): updates and returns current state of guess
3. GetIncorrectGuesses(): returns or prints list of current incorrect guesses
In Driver:

1. `getChoice()`: asks player to choose bow letter or word guess
2. `getGuessL()`: gets the players letter guess
3. `getGuessW()`: "" word guess
Driver: PSEUDO-CODE!

```
int numIncorrect = 0

main(...)

    Hangman game = new Hangman();
    String word = game.selectGameWord();
    println(game.getCurrentWord()); // all dashes
    game.showMan(numIncorrect);
    println(game.getIncorrectGuesses(1, ?));

    while (player can still guess) // numIncorrect < 10 &&
        // player hasn't guessed word yet.
        c = getChoice(); // letter or entire word
        if (c == 'L') // letter guess
            g = getGuessL(); // get letter guess
            if (word contains g)
                (0) println(game.getCurrentWord('?', '?'))
                (1) game.showMan(numIncorrect);
                (2) println(game.getIncorrectGuesses('?'))
            if (all letters guessed)
                player wins!
        else // g not in word
            (1)
            (2)
            (3)
```
General (see hw sheet).
- Game playing in Hangman
- User interaction in Driver
- main() simple

Guideline (see hw sheet)
else // word guess
    w = getGuess();
    if (w correct)
        player wins
    else
        player loses.

Questions:
1- parameter for showman? numincorrect
2- getCurrentWord()
3- getIncorrectGuesses() = your job!

2. getCurrentWord: if guess is in gameWord, adds guess to dispArr

public char[] getCurrentWord(char guess)

//note: don't need if (gameWord.contains(i)) as 2 will detect
for (int i = 0; i < gameWord.length(); i++)
    if (gameWord.charAt(i) == guess)
        dispArr[i] = guess

return dispArr.

May want to have this return a String instead
(for easier printing). Think about how to do that... )