Programming Exercise 6.6

Over-Under Guessing Game, v.1.0

Purpose. Write a simple program from scratch that uses if-logic. The game is played between the computer and a human player.

Supplemental. Read about "randomizing" in http://www.rdb3.com/python/exercises/Gaming.supplemental.pdf.

Requirements. Write **`overUnder1.py`**, the over-under guessing game (http://programarcadegames.com/index.php?chapter=loops), following this algorithm:

Algorithm.

```
1. Call srand.

2. Compute and store the number to be guessed (1-10) using the random
   number generator.

3. Output the computer's challenge to the human, to try to guess the
   randomly selected number.

4. Input and store the human's guess as a whole number.

5. If the human's guess is EQUAL TO the randomly-selected number, output
   the randomly-selected number with a label like "That's right -- it's"

6. If the human's guess is LESS than the randomly-selected number,
   output the randomly-selected number with a label like "That's too low
   -- it's"

7. If the human's guess is GREATER than the randomly-selected number,
   output the randomly-selected number with a label like "That's too
   high -- it's"
```

There are several opportunities for you to consider the human player's experience playing this game, and make it as good of an experience as you can. For example, use labels with proper spelling and spacing. For input prompts, include instructions about what are the possible input values For example, don't just say to enter a guess, instead say to enter a guess in the range 1-10.

Example. User input in blue:

```
I'm thinking of a number between 1 and 10. Guess what it is: 8
That's too high -- it's 6
```