Several early computers games were variations on the user guessing a number randomly generated by the computer. The game of Ricerca Binaria is one variant of this idea.

In Ricerca Binaria, the user tries to guess the computer’s randomly generated number. The computer gives the user hints to tell the user if their guess is too high or too low. See below for an example session of the game.

**Assignment**

Write a program that implements the game of Ricerca Binaria. The program should explain the game to the user. It must pick a random number from a range. In the example, the range is 1 to 100. It must ask the user for guesses until the user guesses correctly. For every guess, the program must give the user a hint about their guess. Finally, the program must tell the user how many guesses they took to guess the number.

**Hints**

You are not required to use the hints. It’s your code.

- Consider separate functions for different functionality in your program. For example, you could make a function to explain the game, another to run the game, and another to give the hints.
- This program has repeated steps that continue as long as the user doesn’t guess the number. What kind of loop is best for that case?

**Potential Sessions**

Sample 1

```
===================================================================
Welcome to Ricerca Binaria!
I will pick a number between 1 and 100, inclusive.
Try to guess my number in as few guesses as possible.
Don't worry, I'm a nice program. I'll give you hints.

Ok, I've picked my number.
Your guess? 30
Too low.
Your guess? 60
Too high.
Your guess? 40
Too low.
Your guess? 50
Too low.
Your guess? 55
That's it!

You guessed my number in 5 tries.
===================================================================
```

**Extra Challenges**

These challenges are ideas for extra features you could add to your game. They are not required for credit, but recommended for fun.

- After the user plays a game, ask them if they want to play again, and repeat if they do.
- Ask the user for their name, and use their name when communicating with them.
- If the user guesses the number in fewer than 6 guesses, give them a complement for a job well done.
- If the user guesses the number in more than 10 guesses, kindly suggest the user could use a better strategy.
- When giving hints, add extra hints of **Hot** if the guess is within 10 of the number, of **Cold** if it is more than 30 from the number.
• Ask the user for the highest number in the range, instead of 100. Be sure to display it in the instructions, and use it in choosing the number.
• If you let the user set the range, don’t let the user give a number that is less than 10, for the range.
• If you let the user set the range, and you are giving complements and strategy suggestions, you need to calculate the when to give them based on the range. Hint: Logarithm base 2 calculations are necessary for this.

Show Off Your Work

To receive credit for this assignment, you must show your source code and demonstrate your running program.

Ask your instructor who they would like you to show the assignment to.