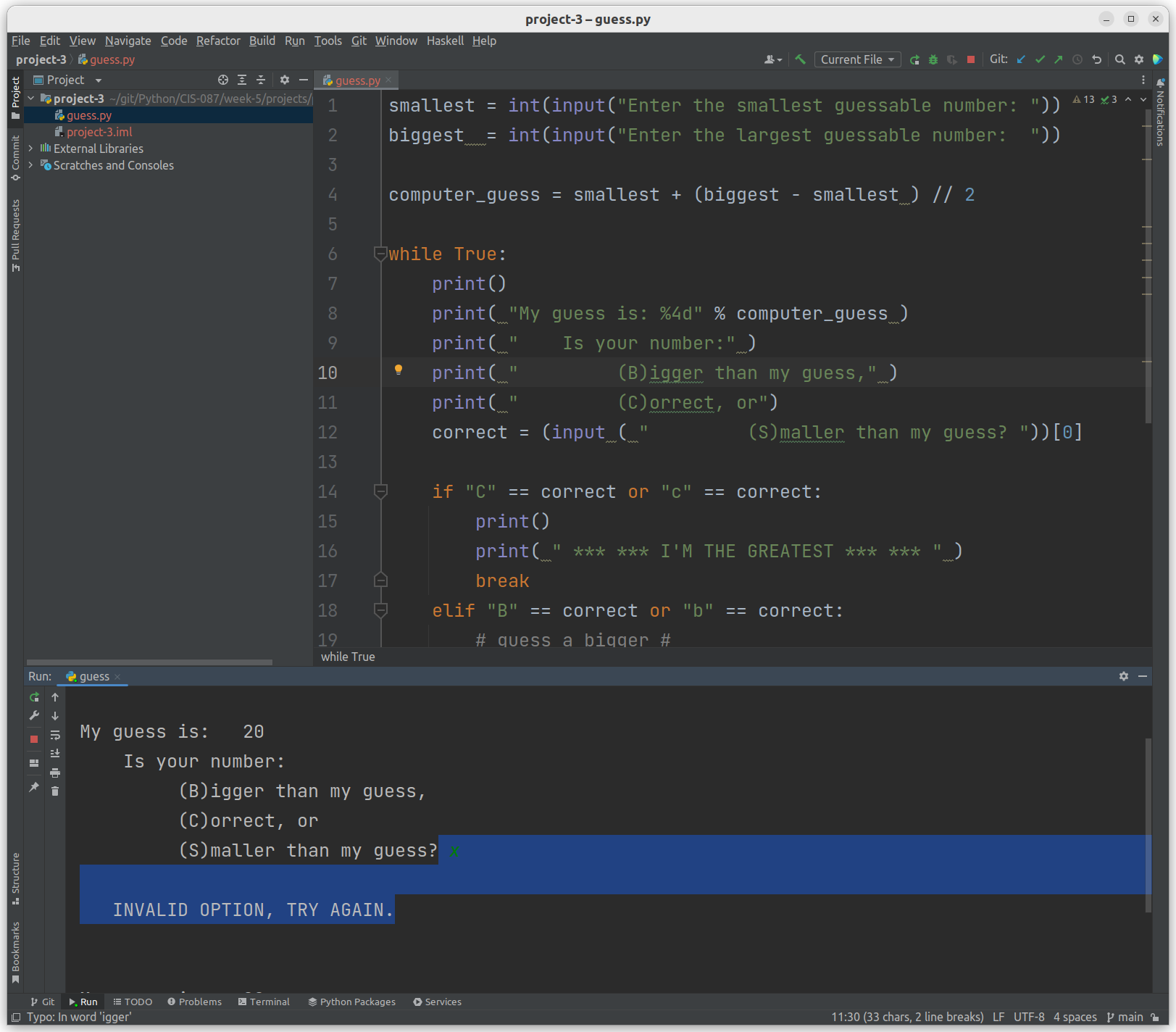
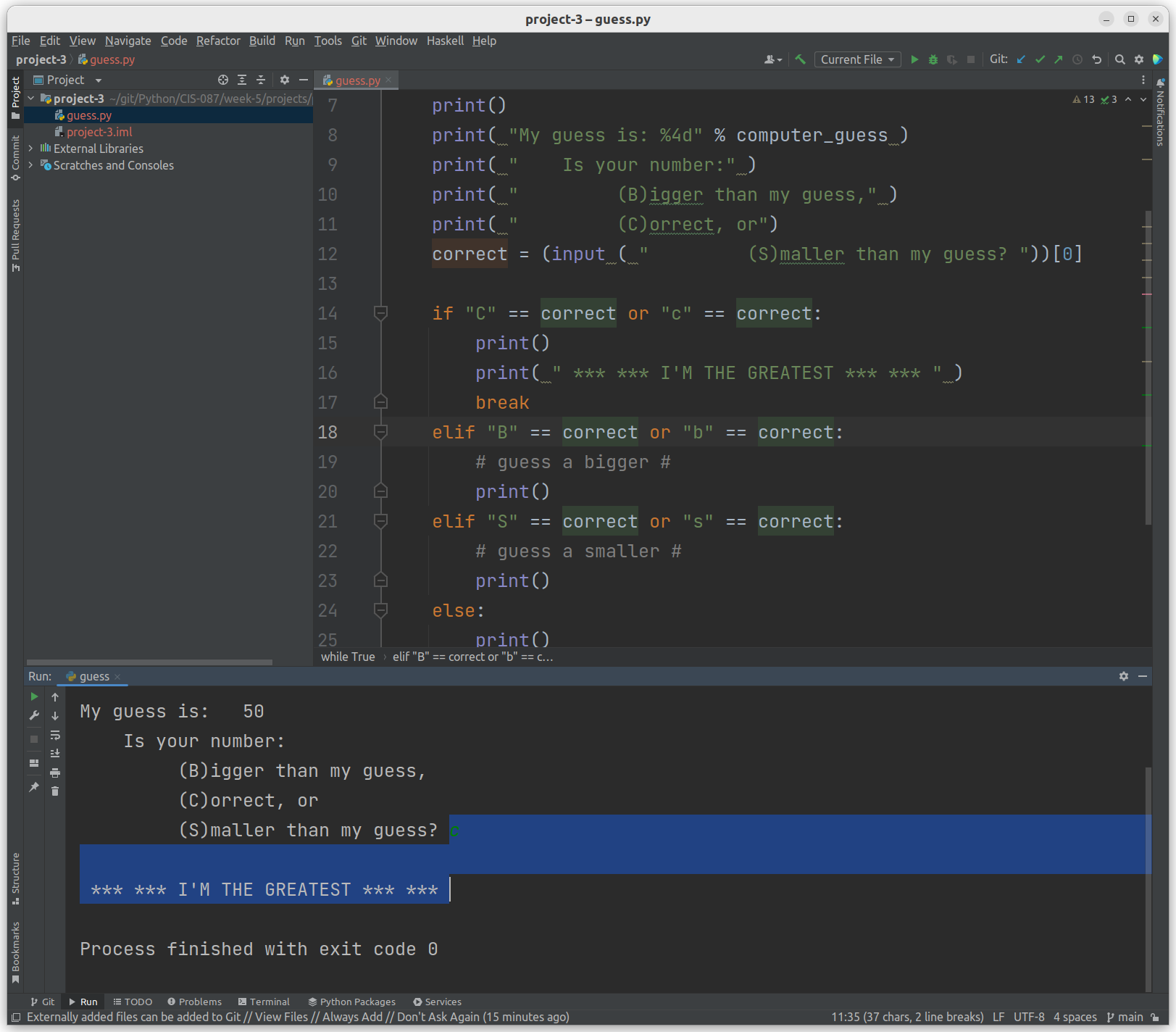
Step 1 = create code skeleton including loop. This is mostly straight from the original guess.py code.

Step 2 = Create if … elif … elif … else loop to handle results of guess.

Step 3 = Test bad input condition.

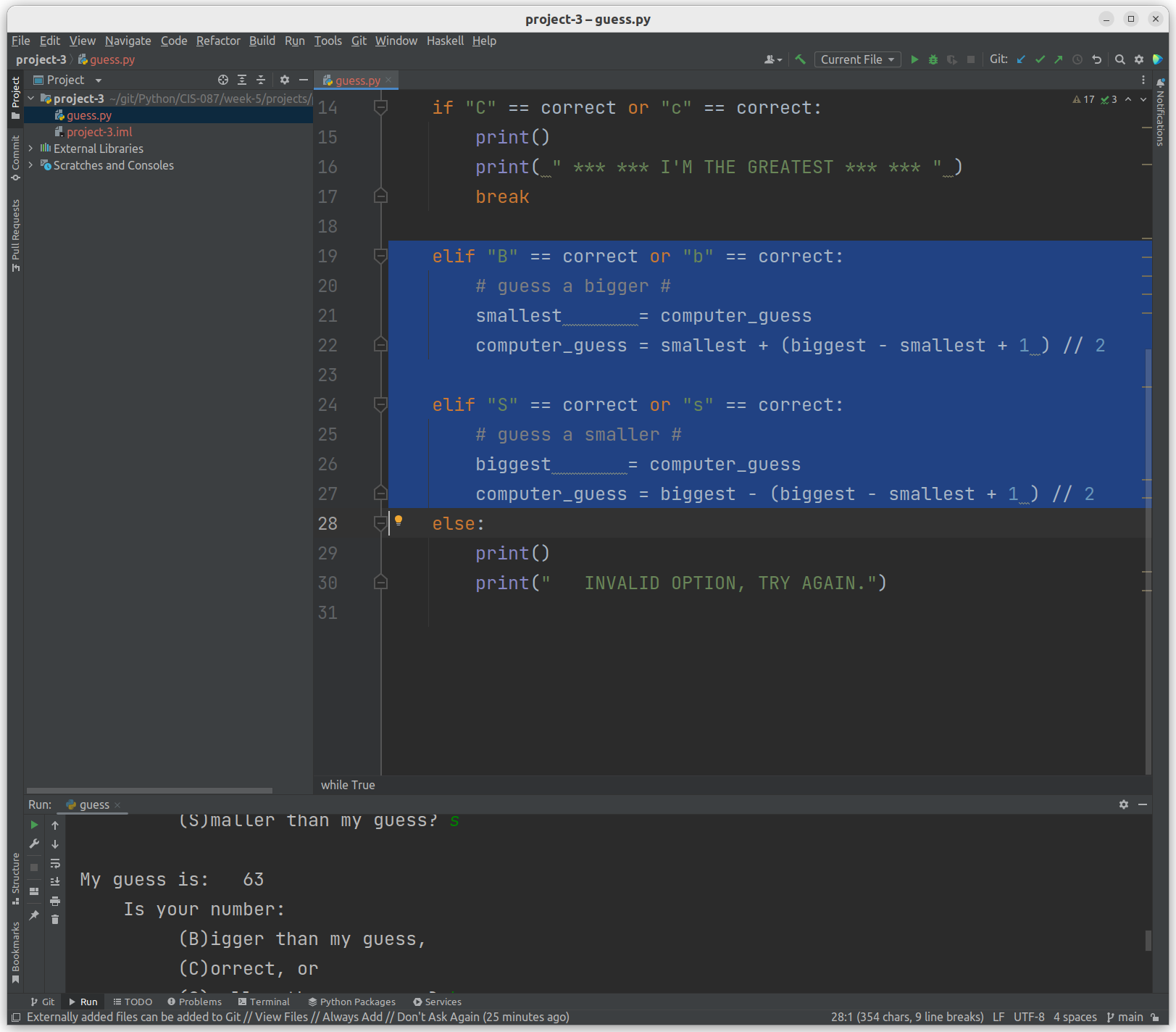


Next: Test Correct condition.

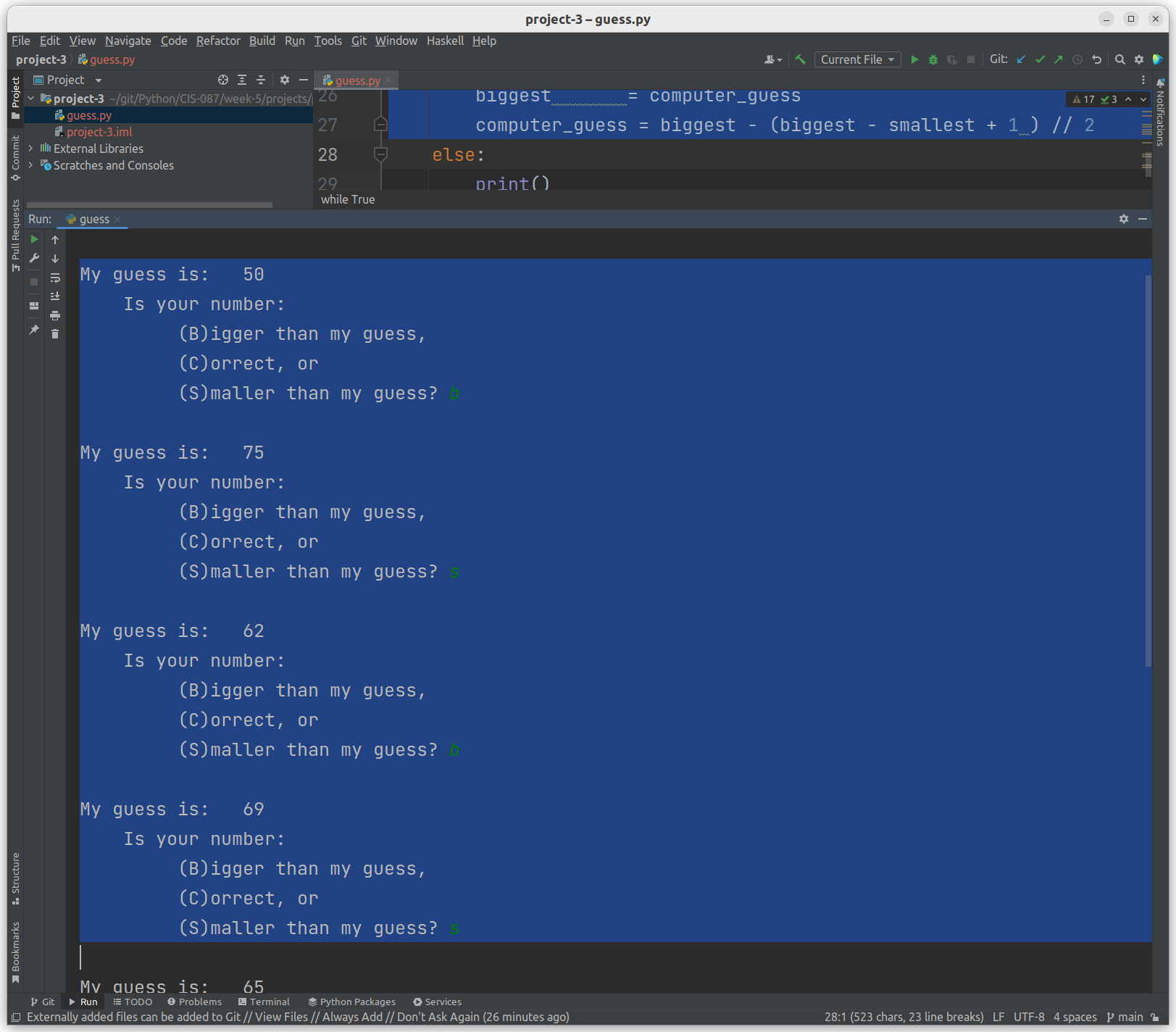


Looking good. Ok, now if the correct answer is bigger or smaller, update our boundaries and compute a new guess. At this point, obviously computer\_guess should be moved into a routine as I am repeating code. However, we aren’t to that chapter, so I’ll do it using brute force.

Code:



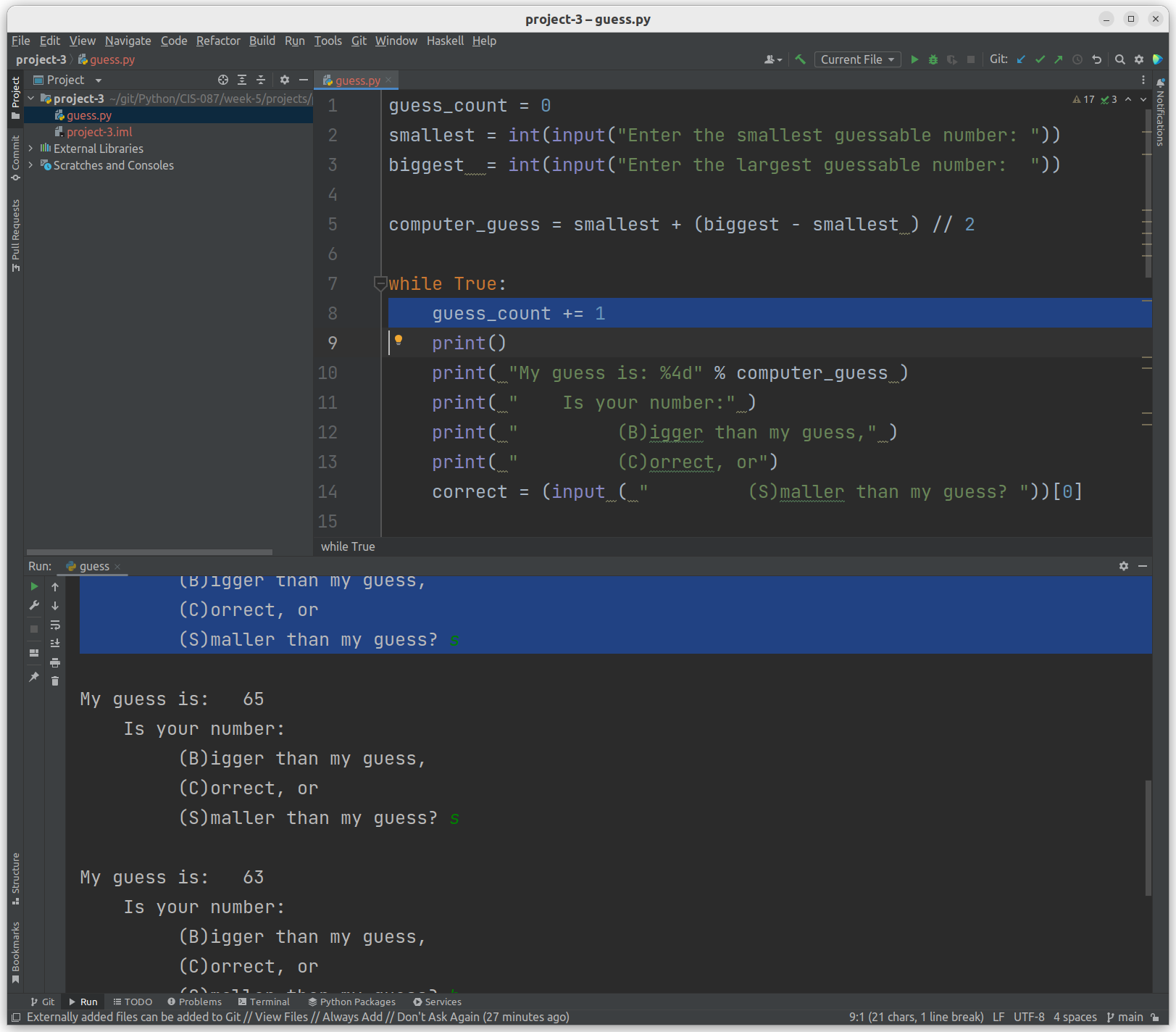
Example Test:



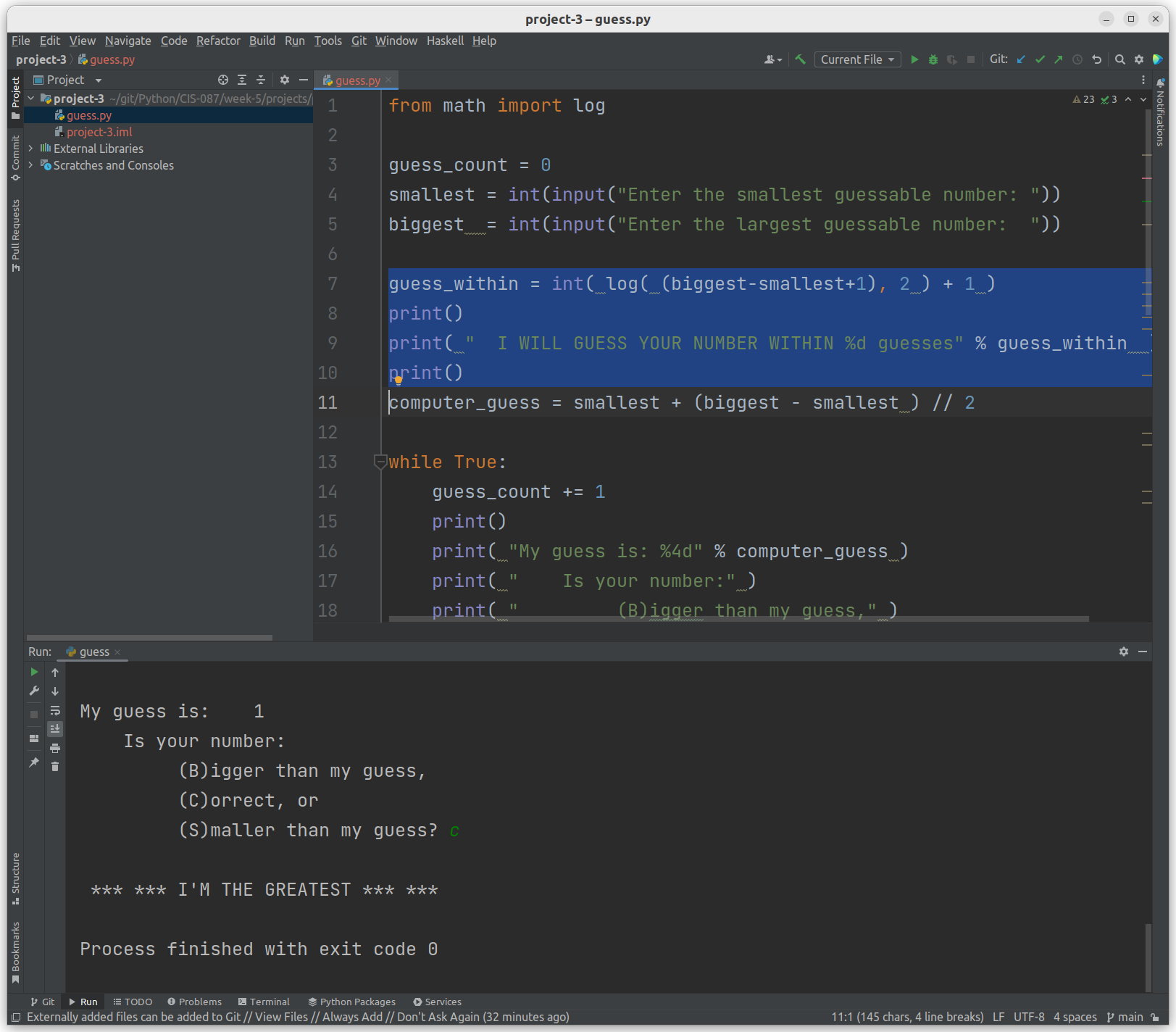
Now for a little fun:

Let’s see if we obtain the answer within the expected time frame:

1) Add a variable (guess\_count) and track the # of guesses with it:



2) Brag about how quickly I can find your number:



3) Announce how quickly I found your number and brag again.

