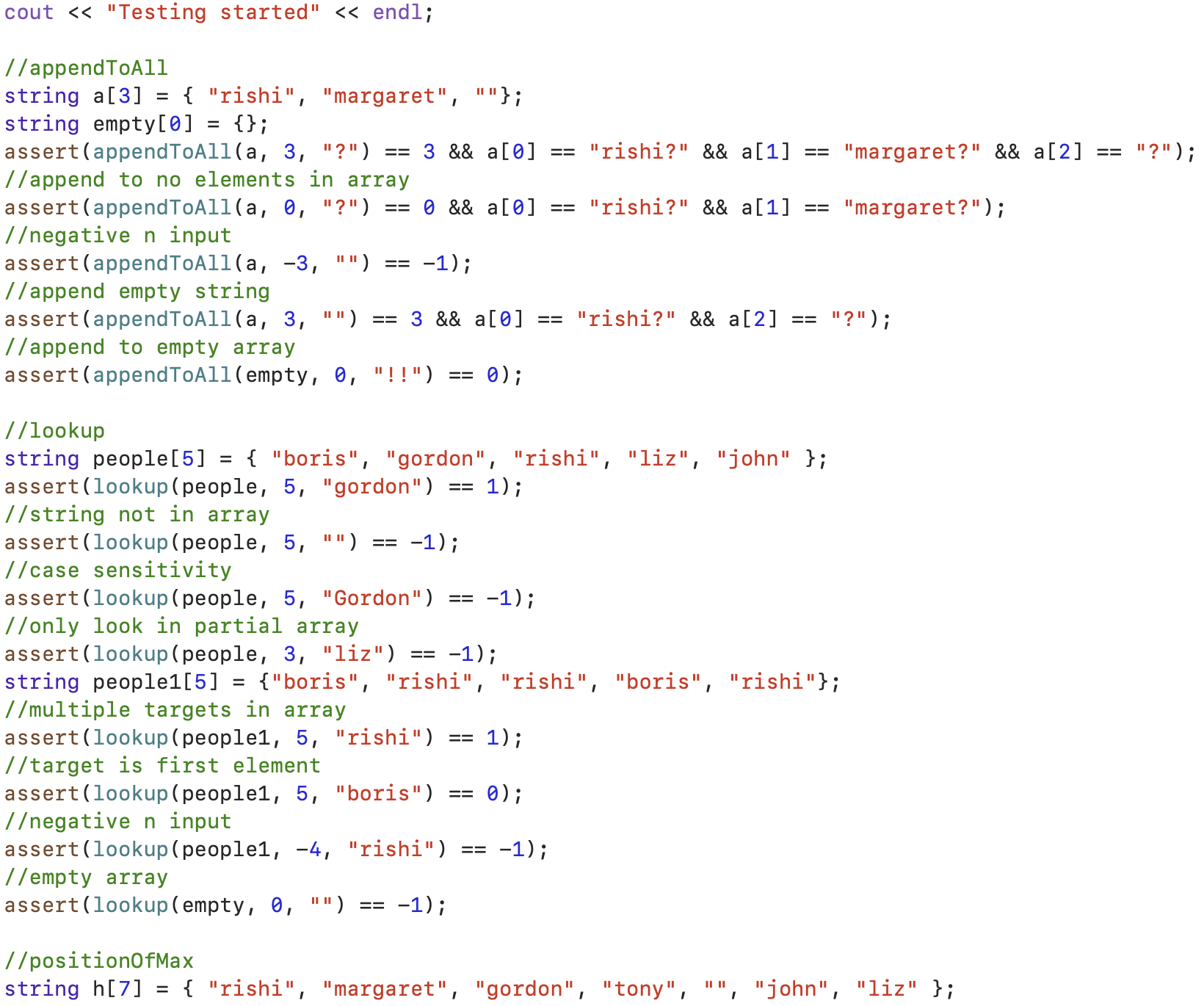
CS31 Project 4 Report – Joyce Chen

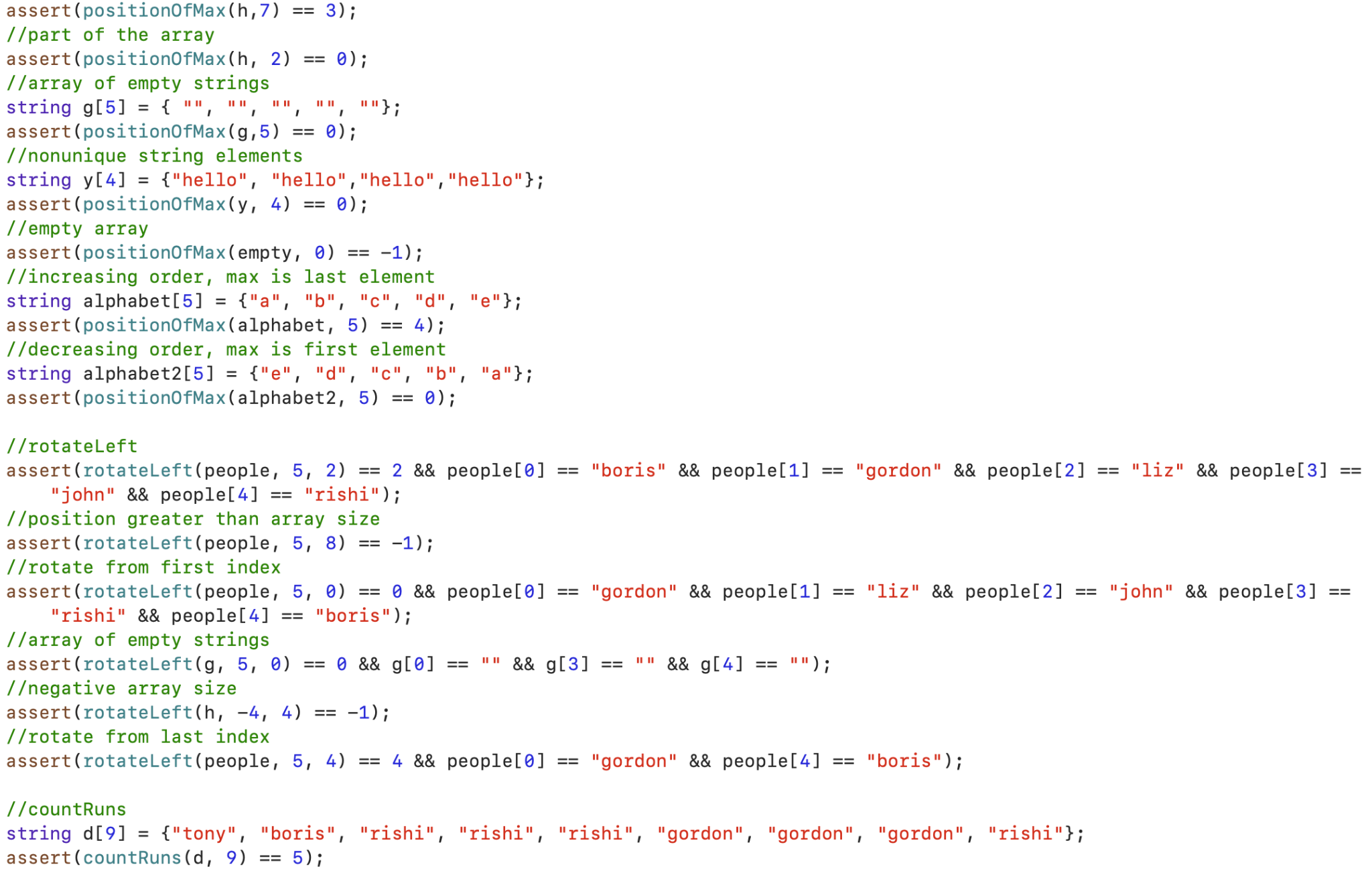
**Notable obstacles**:

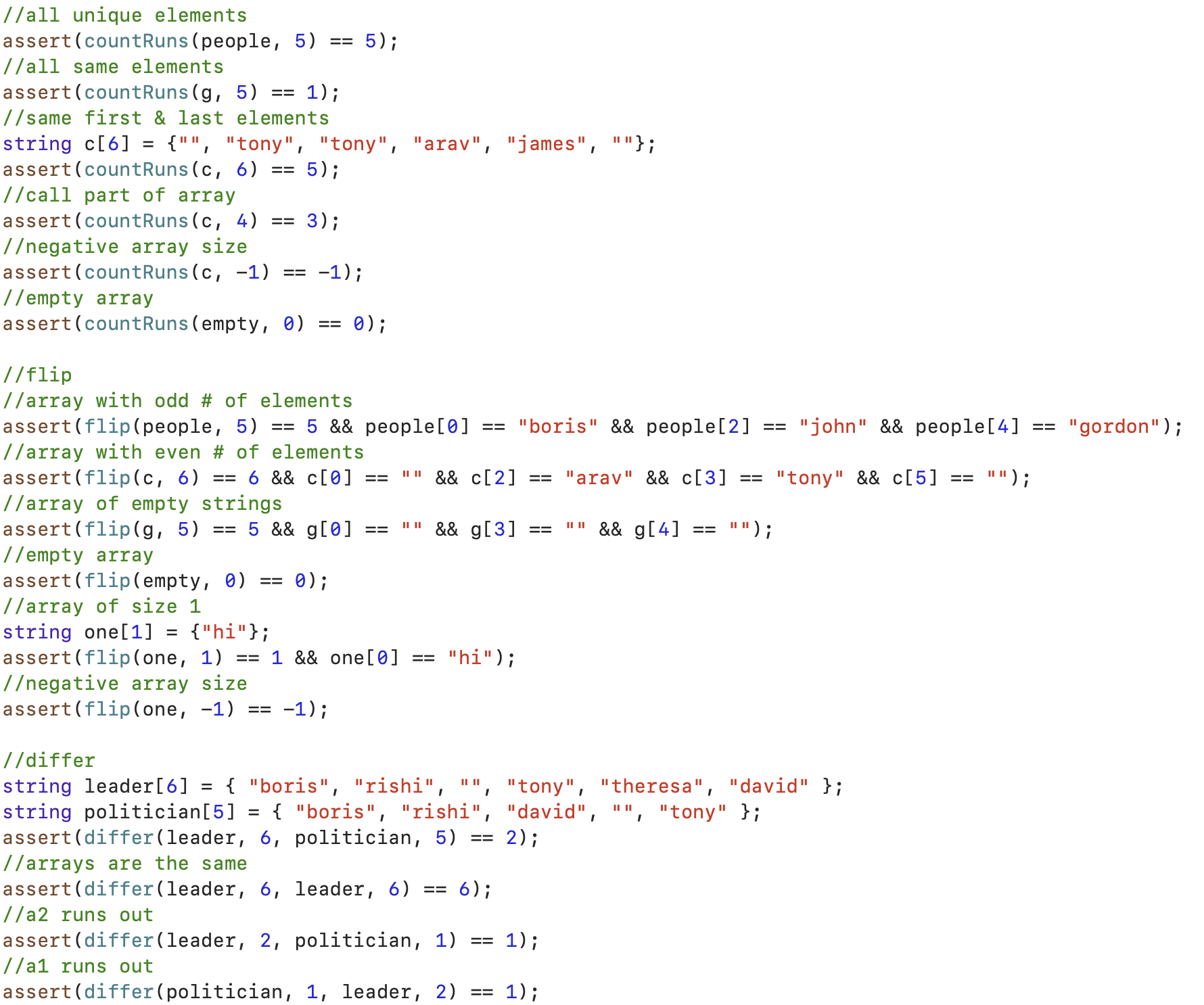
By far the highlight of this project was all its boundary cases. It was challenging to check for invalid inputs in every single function, because each of them were different. For example, while you could pass an empty array into most other functions, you couldn’t for positionOfMax() because there wouldn’t be any interesting elements in the array. For functions like rotateLeft(), you would also have to account for whether the position of rotation would be greater than the size of the array. I spent much of my time in this project thinking of and writing test cases for these boundary conditions. Seriously, life would be so much easier if C++ had an array.length() function, so I won’t have to worry about bad arguments being passed into a function.

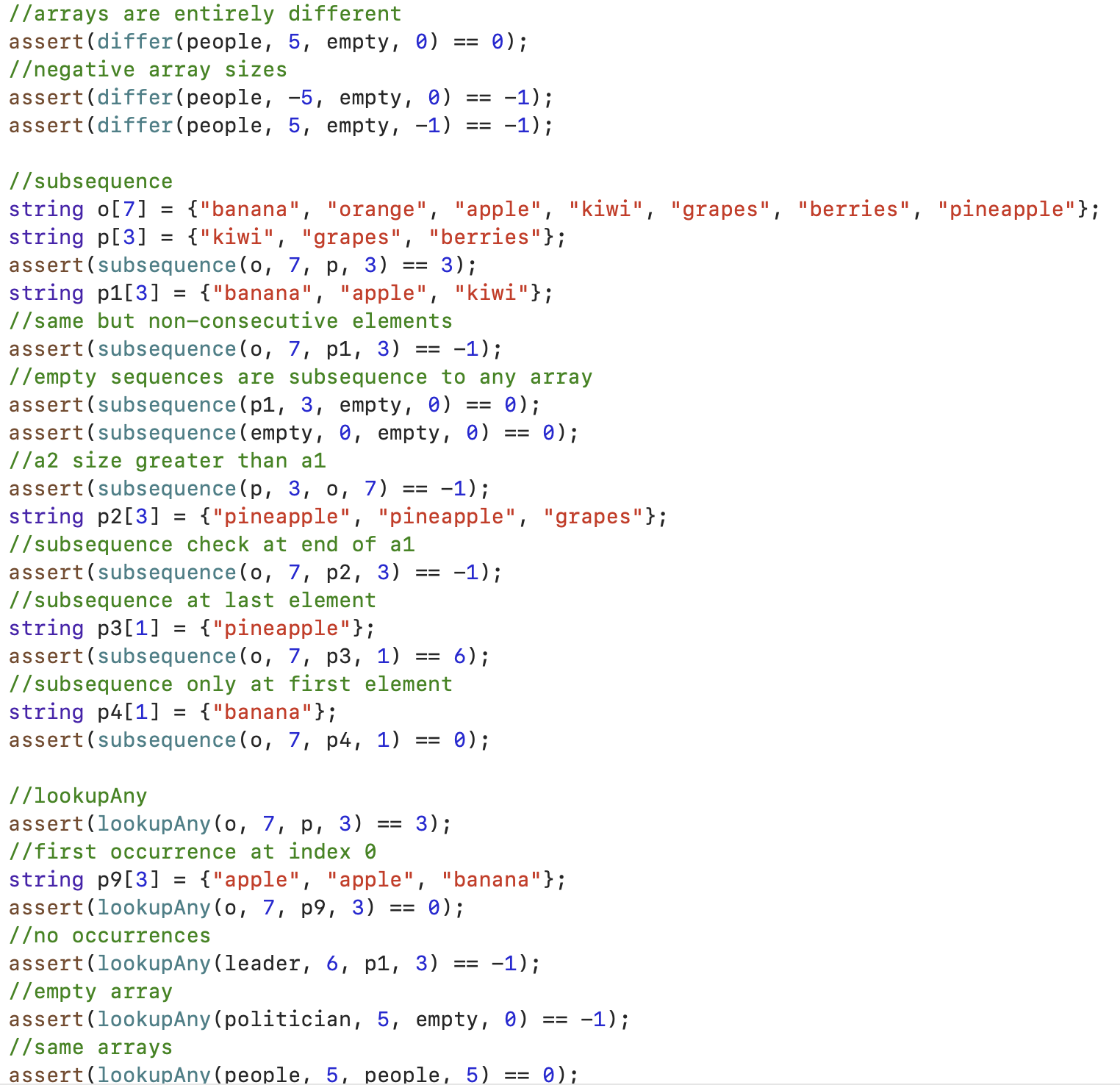
In addition, I found that functions such as subsequence() and lookupAny() were harder to implement, since they required nested loops. The split() function implementation was also difficult to think through – I originally wanted to use the swap function that we learned in the discussion, where we swapped the first element on the left half that was greater than the splitter with the first element on the right half that was less than the splitter. However, after some testing I realized that didn’t work out because the splitter could also be within the array. So I resorted to sorting the array in alphabetical order and then searching for the first element that was greater than or equal to the splitter.

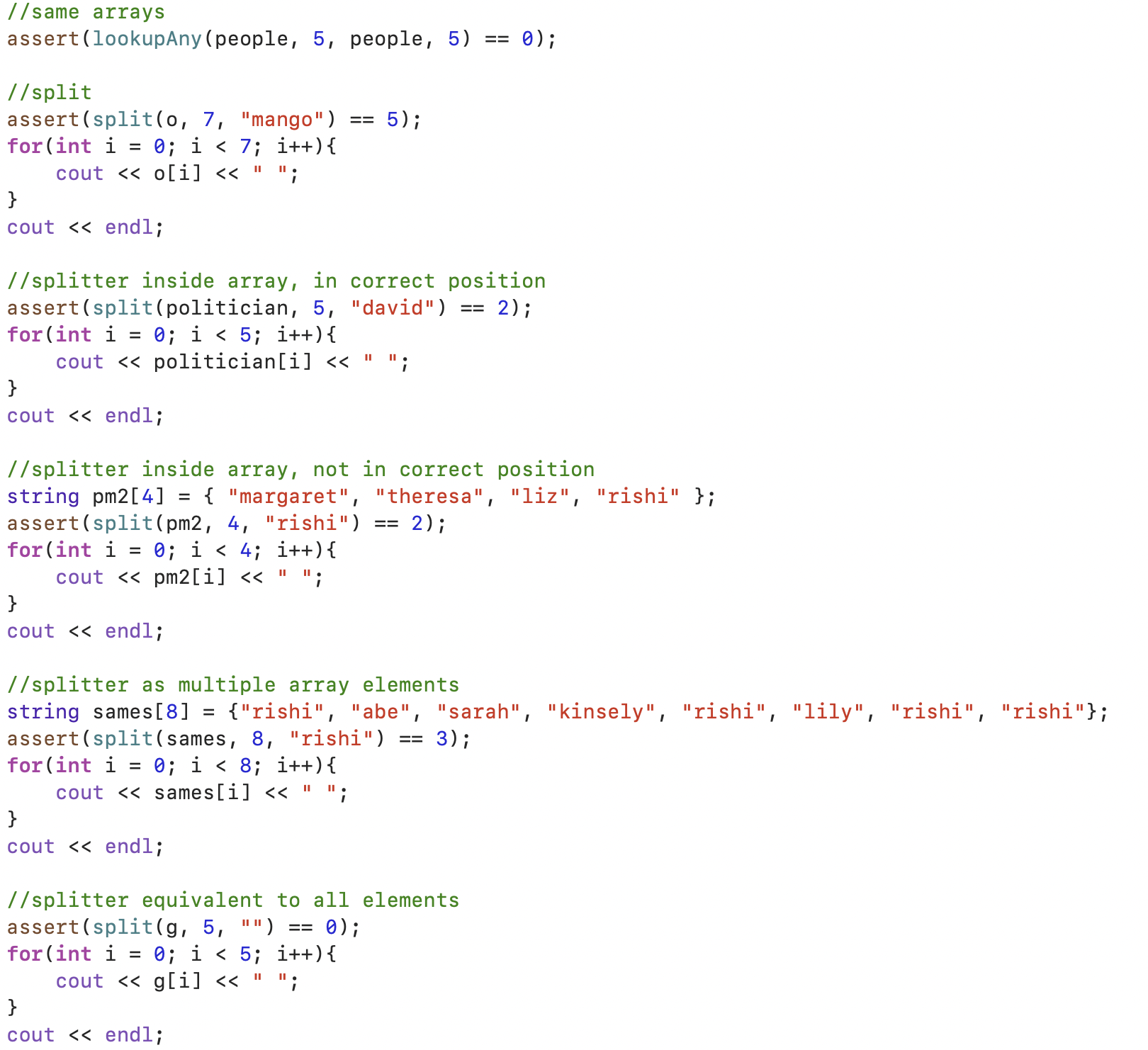
**Test cases:**

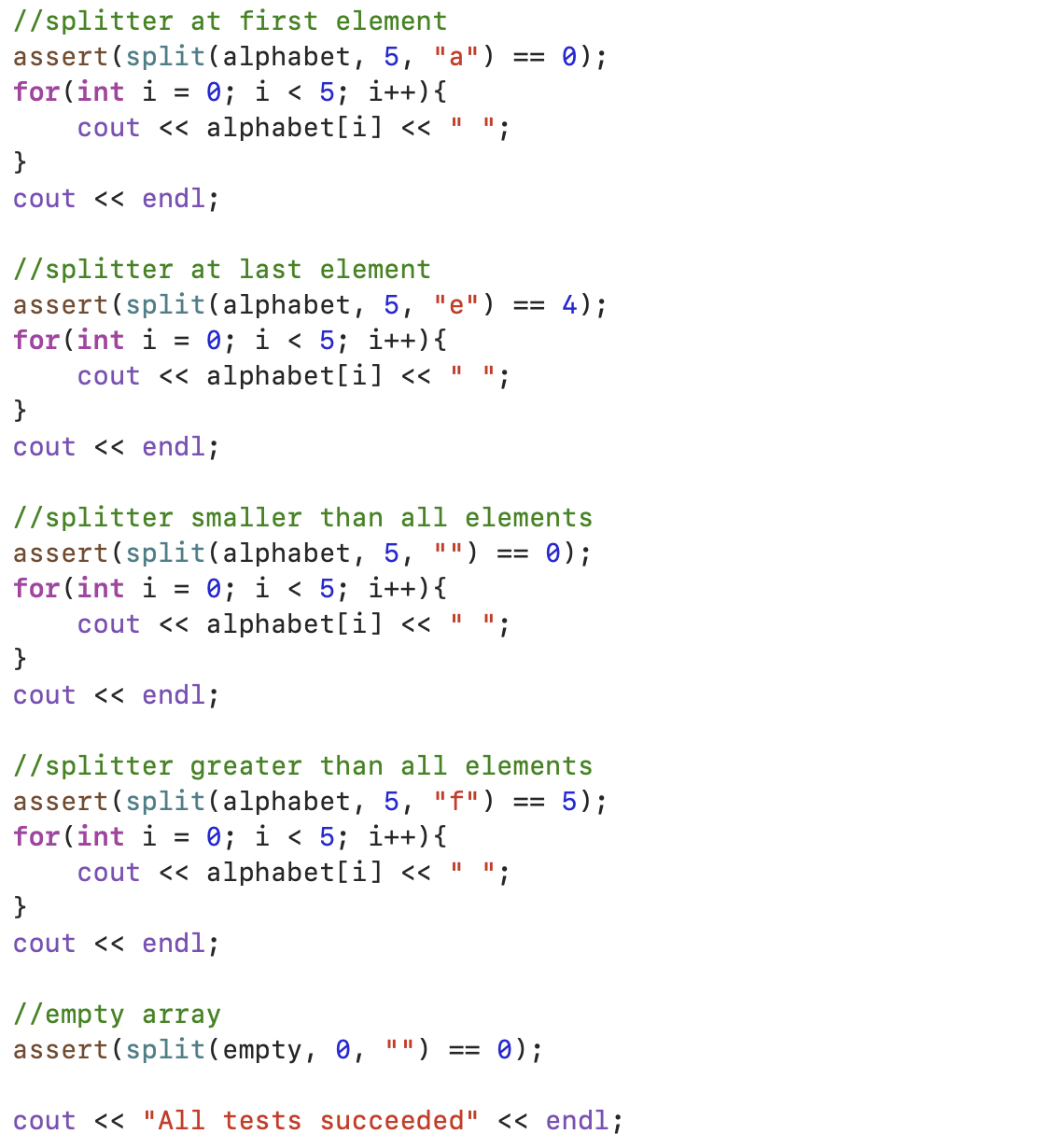
****

****

****

****

****

****