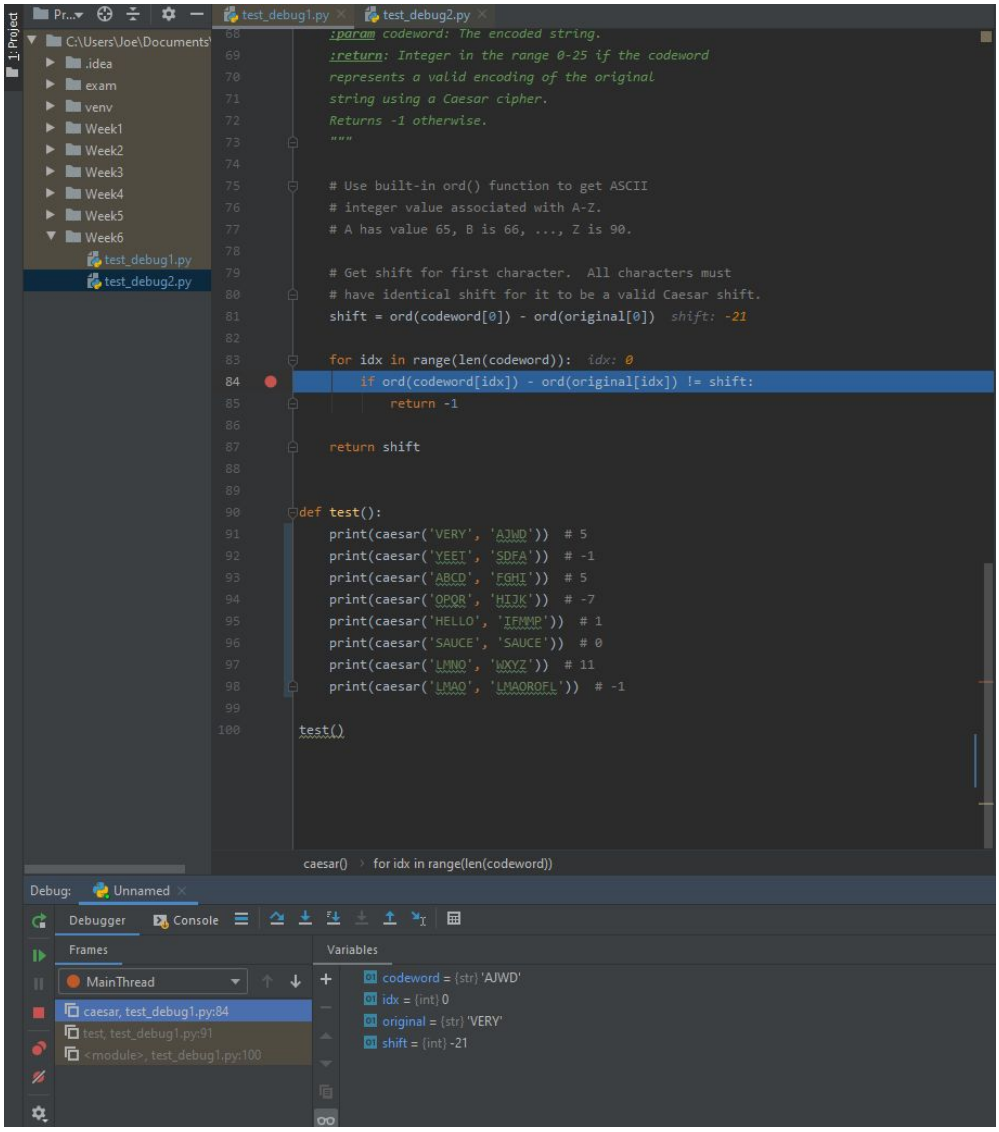
Joseph Vita
CSCI-141 Herring
10/5/19

First Problem

Using the debugger:



Correct Cases

A case that worked despite the various bugs in the code was testing ‘BCDE’ to ‘WXYZ’. It returns a value of 21 which is correct as ‘W’ is 21 characters away from B. Because the code bugs out when the code wraps back around (‘ABCD’ to ‘YZAB’) It works when it is a straight difference in distance. Another case that works by default is when both the codeword and the

original are the same. If they are both the same, the function should return 0 because the strings are 0 ascii characters apart.

Bugs

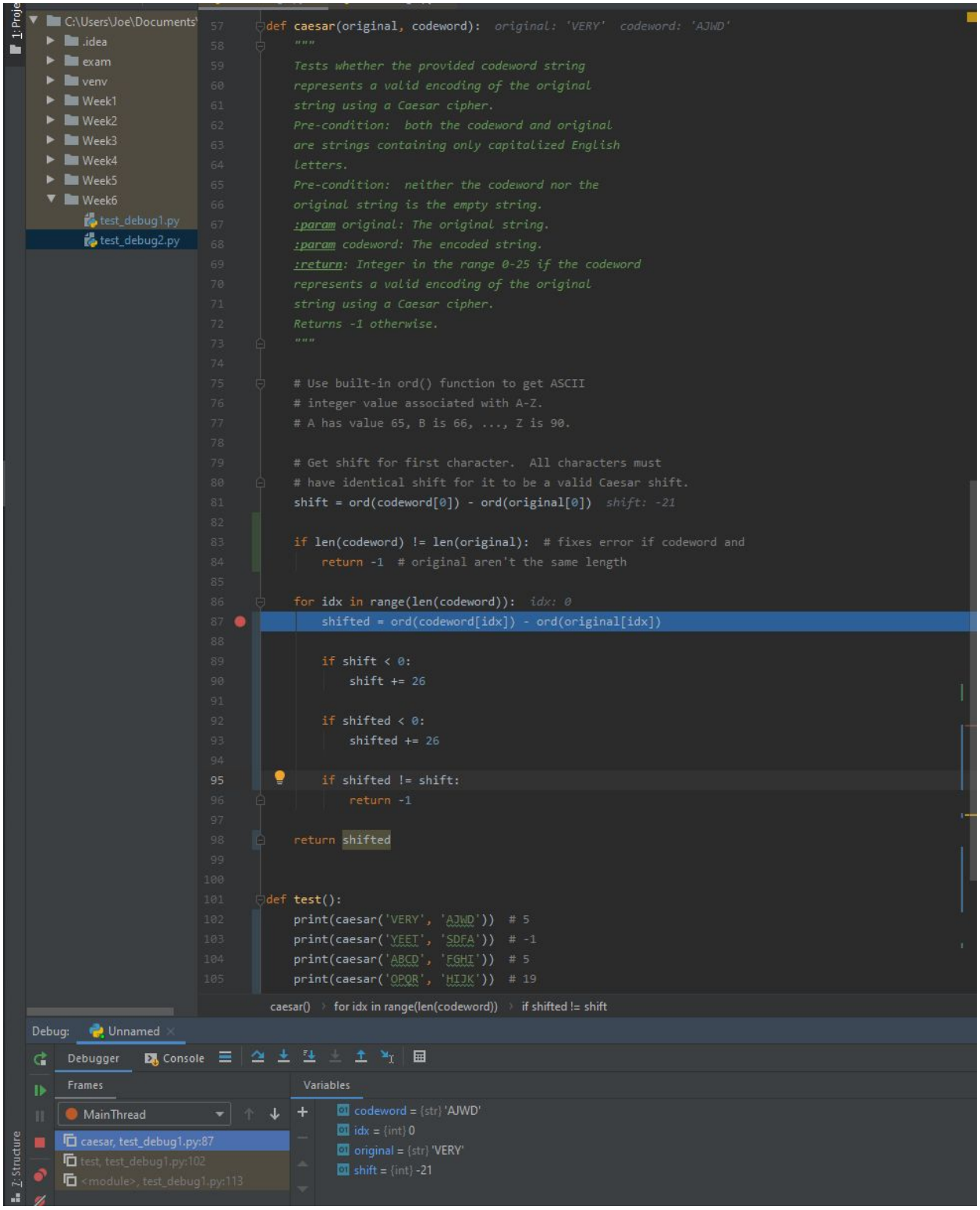
- Program bugs out when strings are different sizes
- Program bugs out when a string ends up going past Z or before A

Fixes

Because the alphabet wraps around to the beginning after z in the caesar cipher, it is possible to end up with a negative number. By adding if statements in the for loop that determine if it wraps around when shift and shifted is less than 0, you can add 26 to shift and shifted to correct the wrap around. Also, I put an if statement out of the loop that just returns -1 if the lengths are not the same. That fixes the different size error.

Second Problem

Using the debugger:



Correct Cases

A case that worked by default was the example case, "established" and "ballistic" which returns 3. The shared string is “lis” which has a length of 3. Also, a case with no matching strings such as "qwertyuiop" and "asdfghjkl" returns the correct answer of 0.

Bugs

- In the while statement, the string1 has a chance to throw an index out of range error if idx1 + this_match_count >= the length of string 1 or idx2 + this_match_count >= string2

Fixes

In the while loop, add an if statement that determines if the program will throw an index out of bounds error and break the loop before it happens