Andrew Plaza

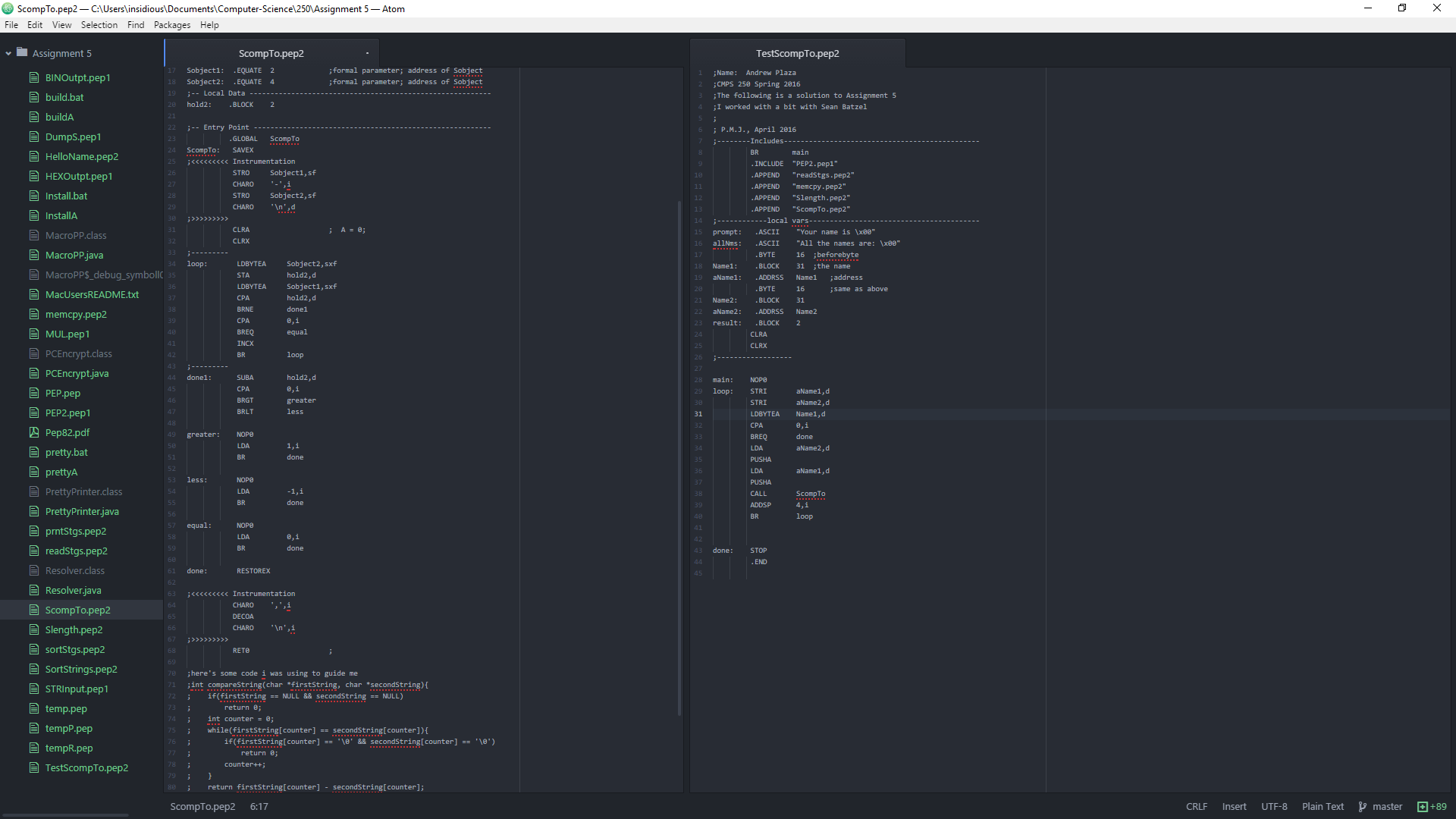
Professor Jackowitz

CMPS 250

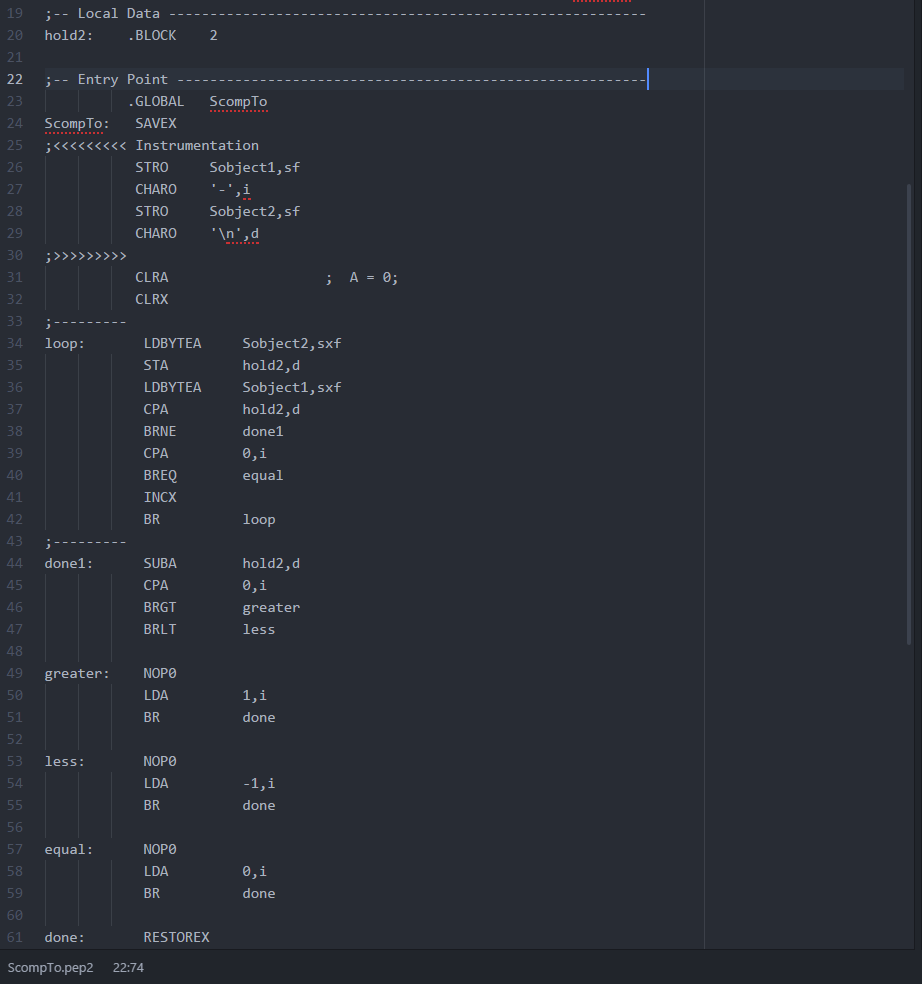
23 April 2016

Assignment 5

**Code In It’s Entirety:**

****

**ScompTo**

**Here is ScompTo (the bulk of the code. Sobject1 and 2 along with the instrumentation below the code is not in this snippet)**

**<make sure the X and A registers are clear for use**

**< Here is the loop which checks every character against the other character in Strings Sobject1 and 2**

**LDBYTEA loads the character stack-indexed deferred**

**<CPA statement checks if the Strings have both reached their nullbyte (x00). If they have reached the end here, that means they are equal so we branch to equal which sets the A reg to 0**

**If they are not equal, one is greater than the other. So we branch to done1**

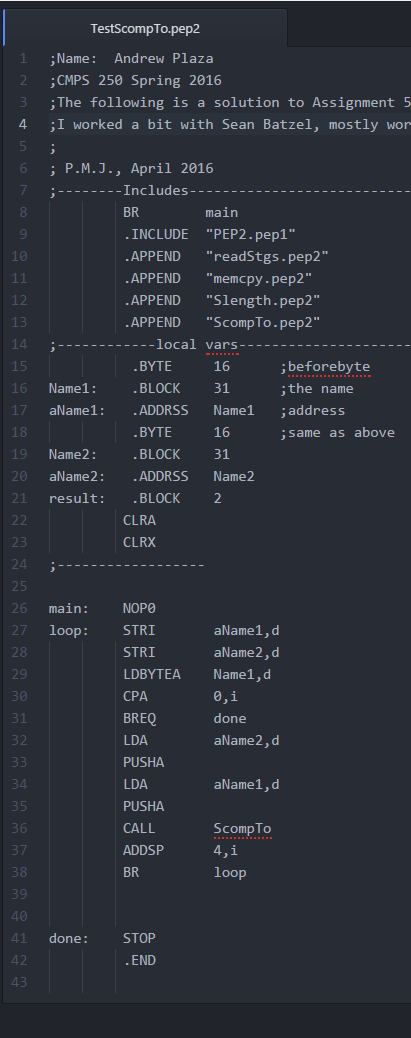
**<done1 ascertains which one is greater and which one is less than by comparing A(char of Sobject1 to hold2(holding the character we are currently on of sObject2)**

**<Branches to less if the char in A is less than the one of hold2, vice versa for greater**

**<If they are equal**

**<Make sure to restore the X register**

**TestScompTo**

****

**<Make sure to get the Includes Right**

**< Create the local variables which will hold the names being read in from the buffer**

**<Result for different debugging purposes**

**<A loop that reads two Strings from the buffer and puts it in the memory location specified by aName**

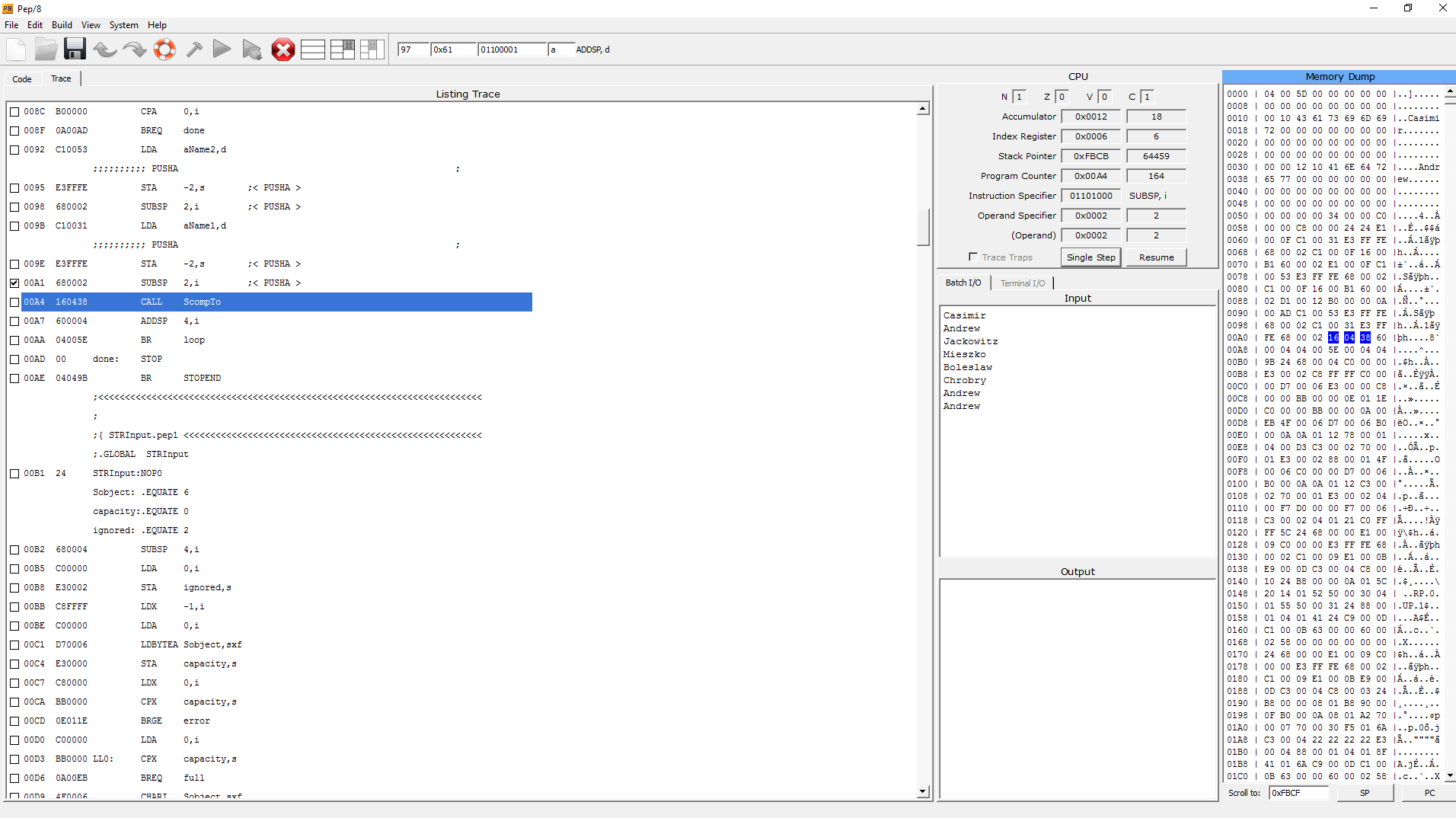
**<Before pushing them onto the stack so that ScompTo can use them, compare one of the names with 0 to ascertain if the empty Strings have been reached**

**<Push the addresses of Name1 and Name2 onto the stack**

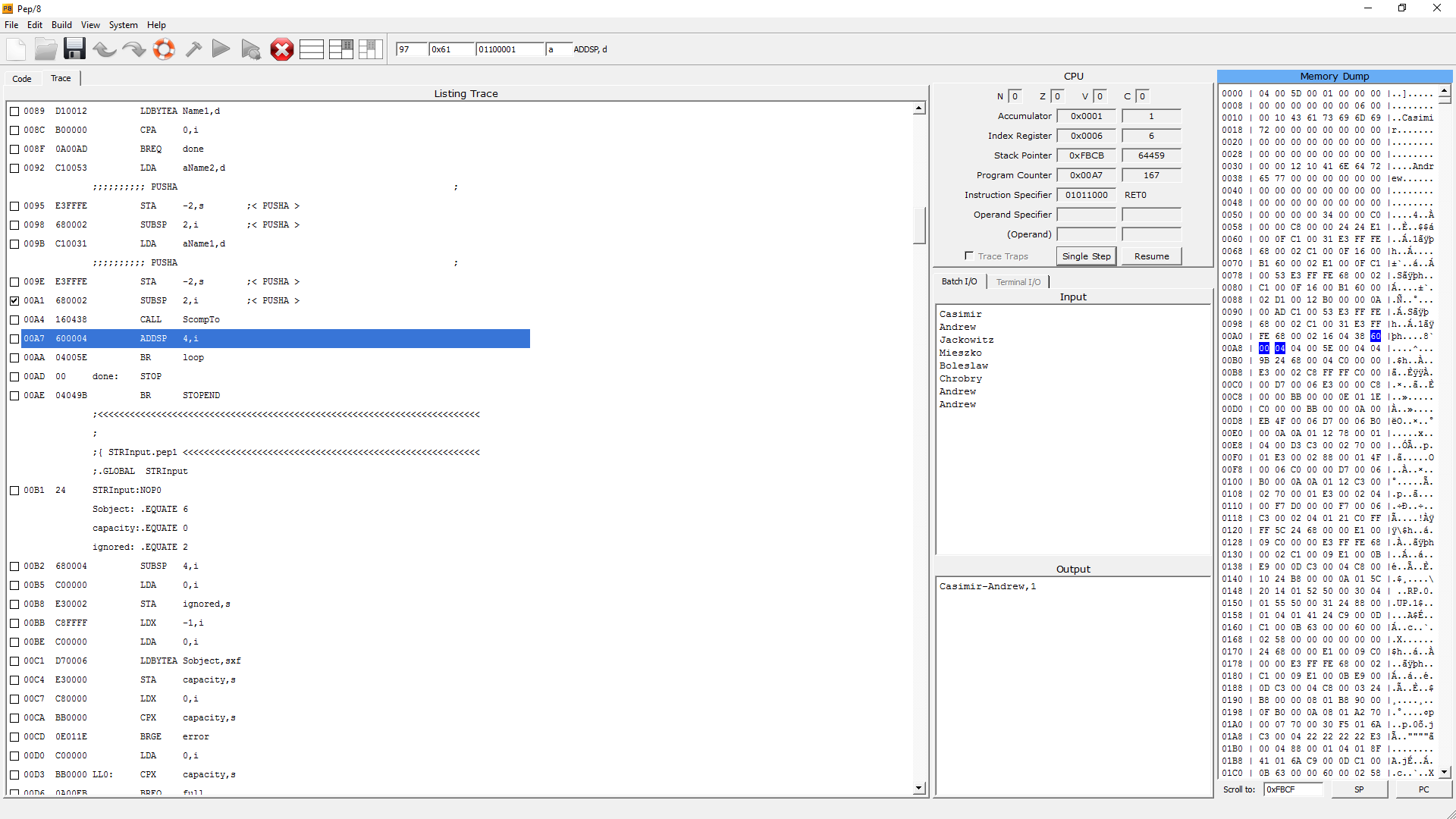
**<call ScompTo**

**<Restore SP to previous spot on the stack (can also be done by RET4 in ScompTo)**

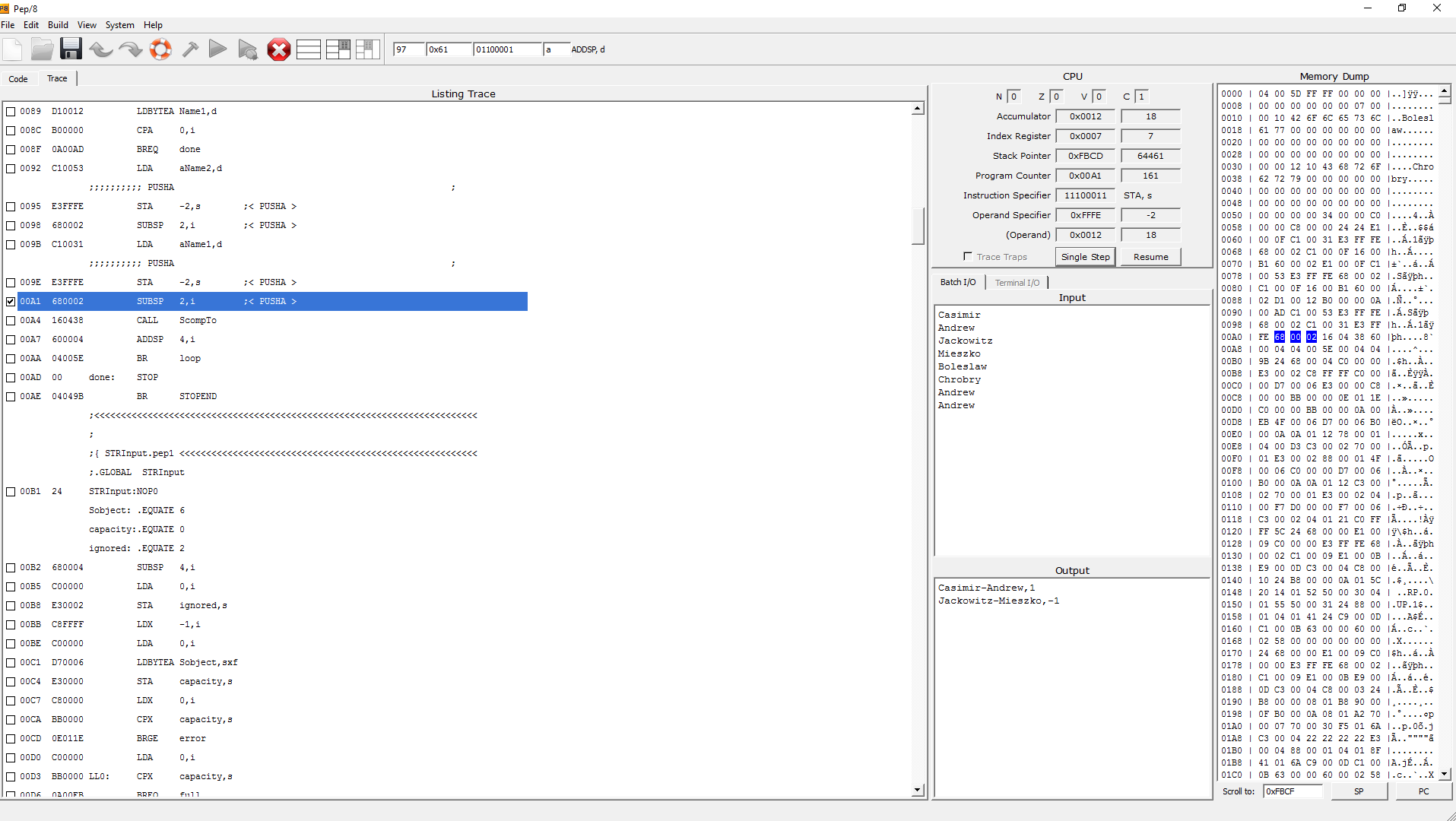
**TESTING**

****

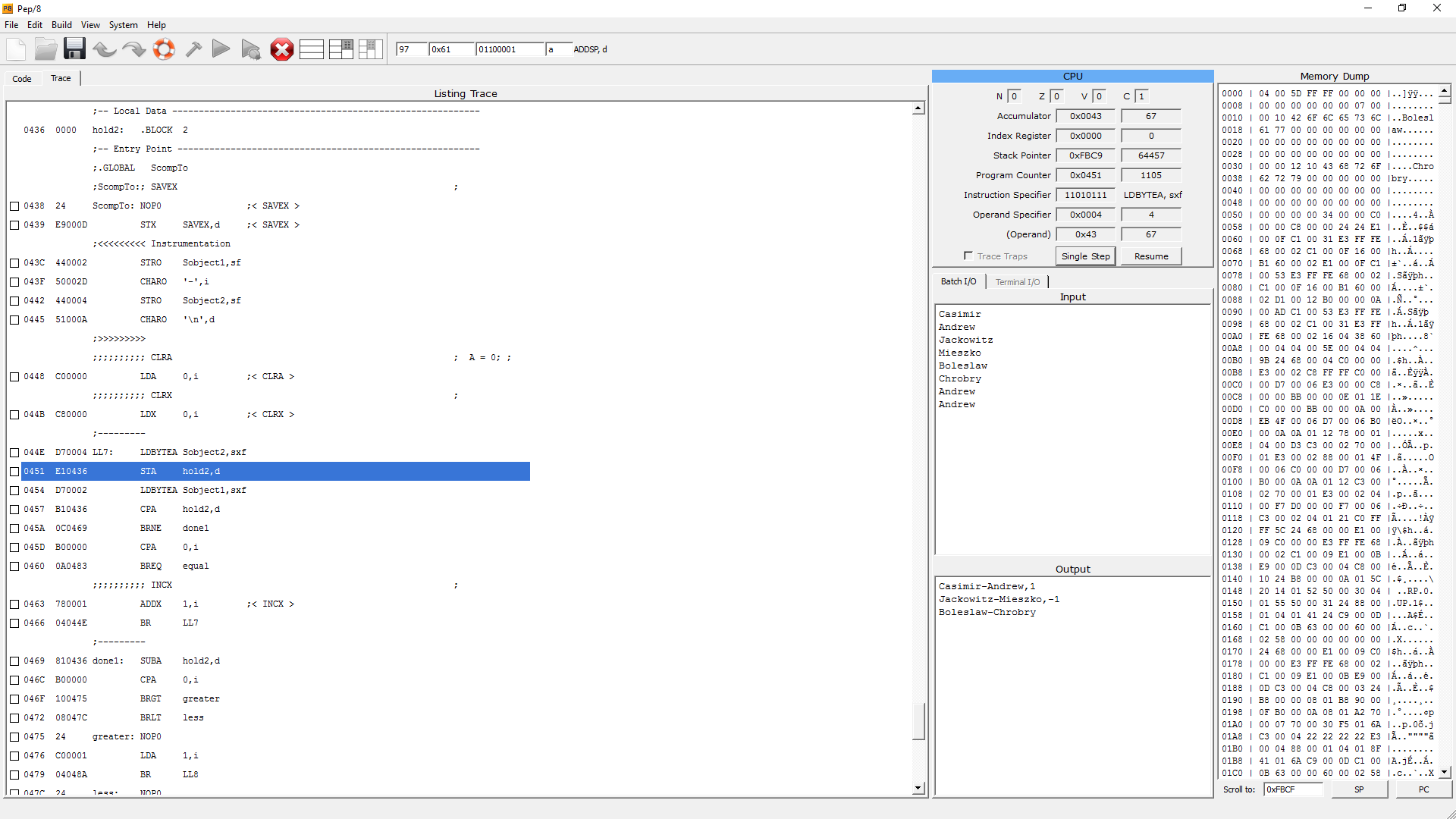
**Calling ScompTo. Casimir and Andrew are already in memory starting at address 0011 for Casimir and 0032 for Andrew**

****

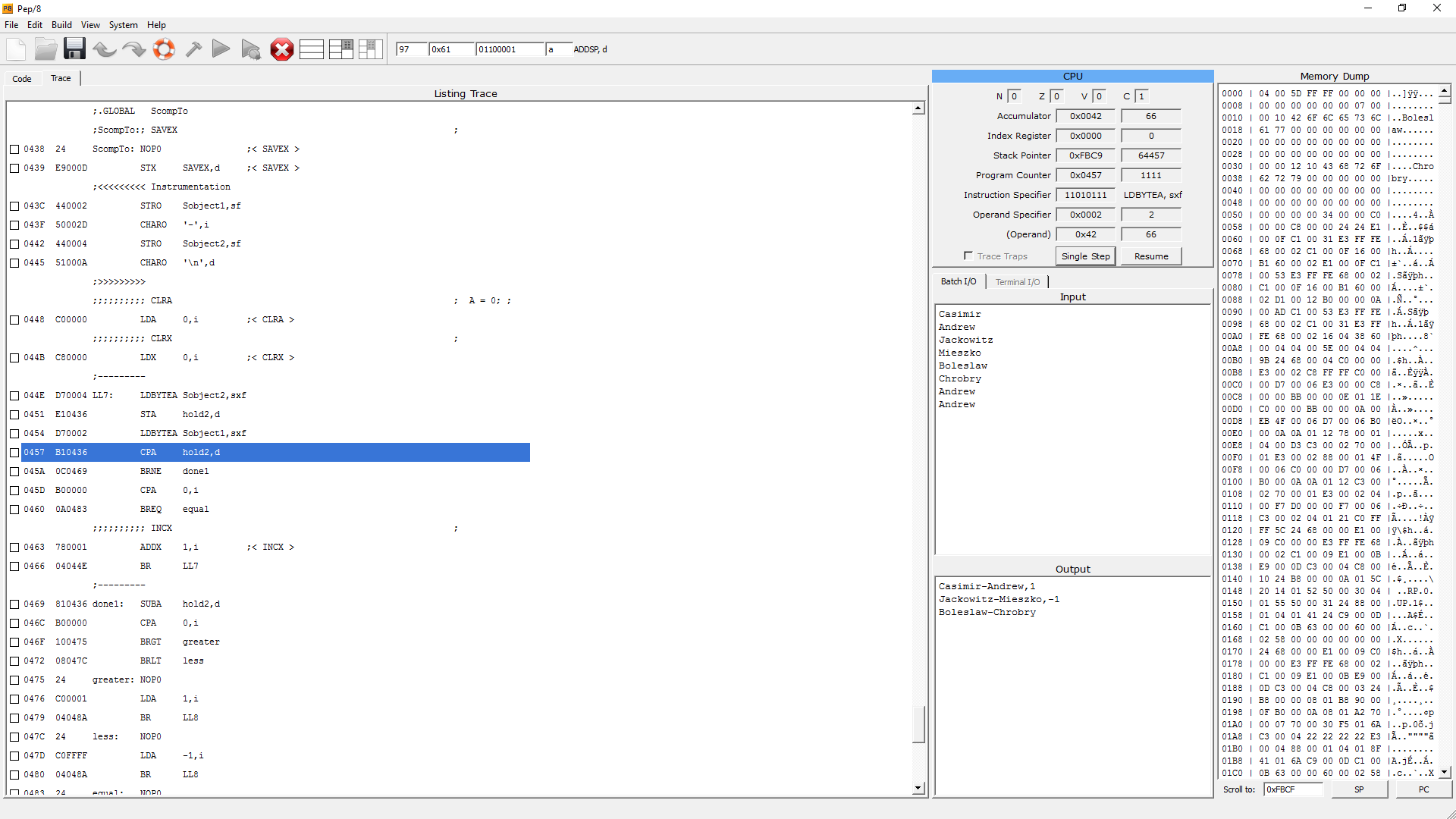
**After ScompTo runs, we see that it returned a 1 in the A register meaning Casimir is lexicographically greater than Andrew**

****

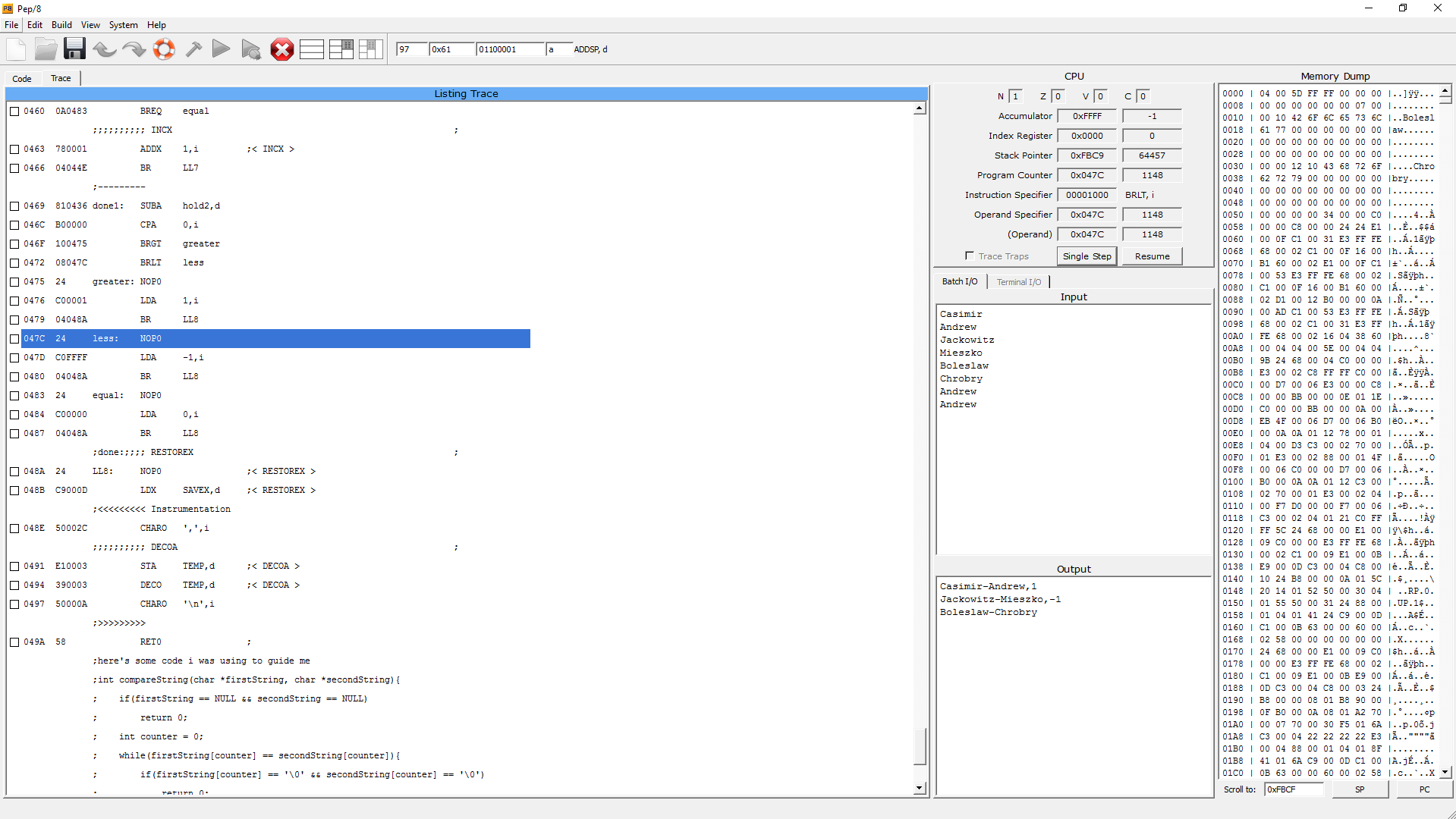
**ScompTo Runs again, this time comparing Jackowitz to Mieszko. It finds that Jackowitz is lexicographically less than Mieszko. At this point in debugging, Boleslaw and Chrobry are in memory queued up to be compared by ScompTo**

****

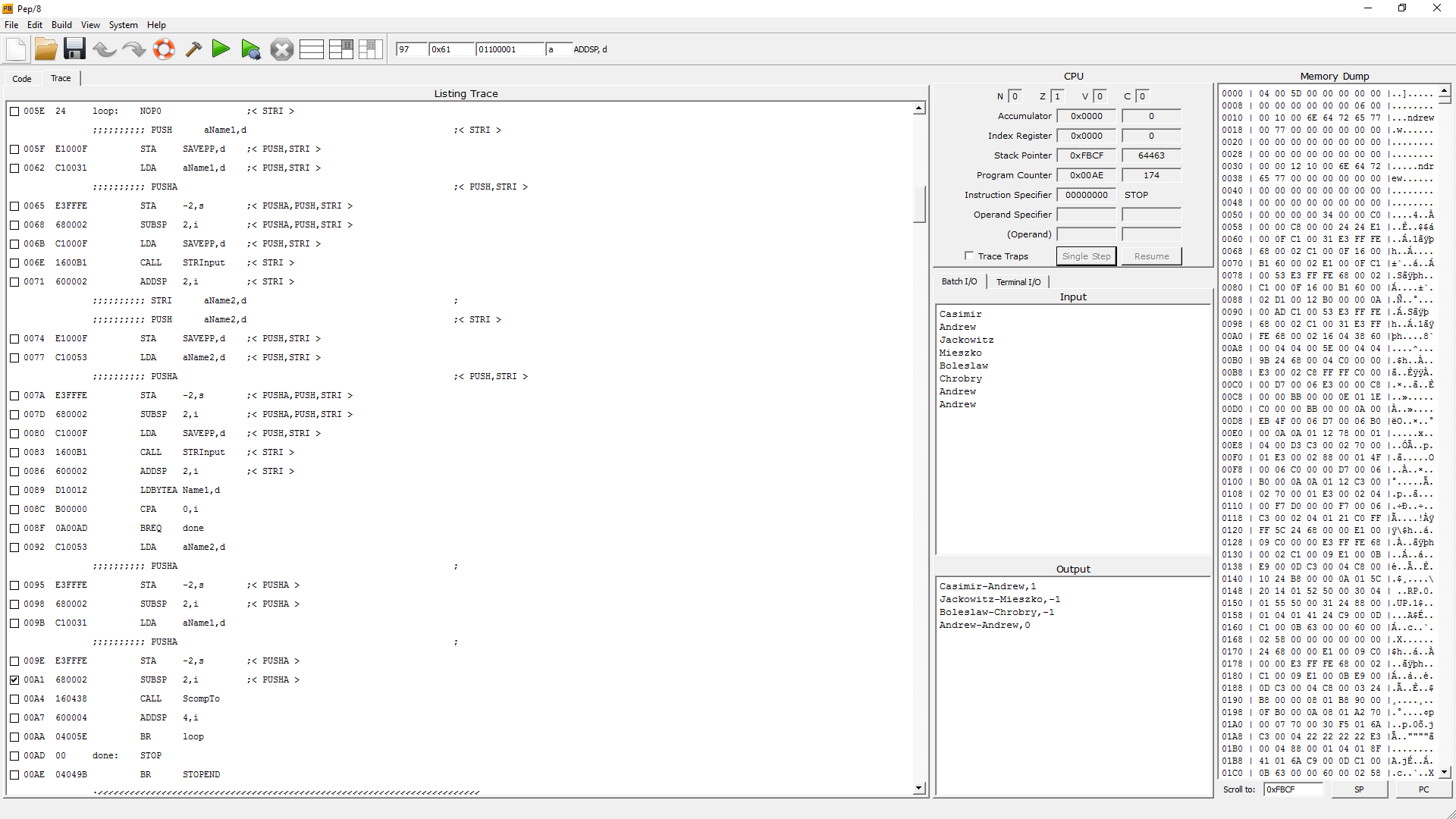
**In the screenshot above (on the previous page) we see ScompTo loading up the first character of Sobject2 (C) in the A register (67 on the ASCII table) and storing it in hold2.**

****

**It then loads the first character of Sobject1 (B, 66 on the ASCII chart) and comparing it against hold2 (67, C). It will find they are not equal to and branch to the done1 method to ascertain which one is greater/less**

****

**The done1 method subtracted hold2 char from A, and found that it was less than 0. Therefore we Branch to the less method to set the A register to -1**

**In this screenshot, we have finished the subprogram. Boleslaw-Chrobry got a -1 just like we would expect, and Andrew-Andrew, two equal strings, got a 0 just like we would expect. Execution then finished since TestScompTo came across an empty String. In all, the program runs as expected and is a solution to Assignment 5 Part I**