1.5 Supervising the computer lab

This puzzle is a program written in the programming language Shakespeare. A description of the language can be found at

`http://shakespearelang.sourceforge.net/report/shakespeare/shakespeare.html`

In this language, various nouns and adjectives are assigned positive, negative, and neutral values. A list of these can be found here.

`http://en.wikipedia.org/wiki/Shakespeare_(programming_language)`

First, three variables are declared: Olivia, Orsino, and Viola. In Act I, Scene I, Viola is assigned the value 64, and then is set to 80. We also assign Orsino the value 15. In Act I, Scene II, Olivia remembers a stack. From bottom to top, Olivia is the stack

$$15, 7, 4, 15, 2, -2, -5, -4, 15, -3, 80.$$

Finally, Olivia speaks and sets Viola = 2 and then Viola = 10.

In Act II, Scene I, Orsino = 80, popping the 80 off of Olivia's stack. In Act II, Scene II, Olivia pops the top value from the stack (let's call it x), computes Orsino $- x$ (that is, $80 - x$), and outputs the character corresponding to that number in ASCII. Then Viola decreases by 1, and (provided that Viola is a positive number) we loop back to the beginning of the scene.

This results in outputting the ASCII characters corresponding to the numbers 83, 65,84,85,82,78,65,76,73,65. These spell the answer: SATURNALIA.

Remark: A compiler for the Shakespeare programming language — or more precisely, a package for converting a Shakespeare program to a C program — is available at the sourceforge repository listed above. Unfortunately, a bug prevents this package from compiling.

In fact there is a stackoverflow question about this issue, and the first answer explains how to fix the problem. (Google "stackoverflow compiling and executing the shakespeare programming language" to find this.) After fixing the bug, with a bit more effort is then possible to compile and execute the puzzle to obtain the answer.