You are going to modify a program that I wrote in three ways. The program is in a single file that I named "sillyCode\_given.c". As provided, the program consists of the required "main" function, a function called "f1" that is called once, and a function called "hello" that isn't called at all. This program is silly; it does nothing at all useful. All it does is display the memory addresses of certain global and local variables and parameters; when it gets back to main, it displays the values of certain global and local variables.

As is, the program should compile with gcc and produce semi-similar results on any system. The exact memory address that you see will almost certainly be different (they will likely even change from run to run) This is the output I saw during one run on an Ubuntu virtual machine:

In main, before calling f1:

gi1 is at address 0x55da2be00010

gc1 is at address 0x55da2be00014

li1 is at address 0x7fffc87aa564

lc1 is at address 0x7fffc87aa563

In f1:

gi1 is at address 0x55da2be00010

gc1 is at address 0x55da2be00014

li1 is at address 0x7fffc87aa53c

lc1 is at address 0x7fffc87aa53b

lvp1 is at address 0x7fffc87aa540

pi1 is at address 0x7fffc87aa52c

pc1 is at address 0x7fffc87aa528

In main, after calling f1:

gi1 is at address 0x55da2be00010

gc1 is at address 0x55da2be00014

li1 is at address 0x7fffc87aa564

lc1 is at address 0x7fffc87aa563

gi1 has value 200

gc1 has value Z

li1 has value 50

lc1 has value C

You will add code near the bottom of the f1 function where you see the comment stating the following message: /\* YOUR CODE GOES HERE \*/

I am asking you to create *three modified versions* of this program that act as follows:

* Version 1: Your code should change the values of gi1 to 150, gc1 to 'Y', li1 to 25, li1 to 25, and lc1 to 'B' (but there is a catch, to be explained shortly).
* Version 2: Your code should modify the return address to point to the hello function. So, when f1 returns, the message "Hello World!" will be displayed, and the program will end.
* Version 3: Your code should modify the return address so that it jumps back to main, but one instruction earlier than it would have, so that f1 is called again. Note that this will happen repeatedly, but at the start of the fifth call to f1, the program will end (due to existing code).

*You must adhere to the following constraints*: The only variable that you are allowed to modify in your code is lvp1, a local void pointer in function f1. To the right of assignment operators, the only other variable you are allowed to access is the global variable g1.

For versions 2 and 3, you are allowed to call the following function: \_\_builtin\_return\_address(0). I was not previously familiar with this function; I searched for a way to do this when I was coming up with this assignment. This function, called this way, returns the memory address that is the return address of the current function. However, it does not tell you where this is stored in memory; you will have to search for that! You can do this by making lpv1 point to itself, and then searching nearby (probably in higher memory) to find the return address; then, you can modify it. (Technically speaking, there is a small chance that this won't work, if the same value exists somewhere else in memory by coincidence; however, that is highly unlikely, unless you do something wrong.) To change the return address to refer to the entry point of another function (such as the hello function, for version 2), you can simply assign the desired target function (by name) to the memory where the return address is stored.

Note that parts of your code may need to differ between operating systems and compilers. Therefore, when you submit your code to me (three modified versions of the program), also include screen captures of the outputs of each version.

The first thing you should do is to step through my code, and make sure you completely understand what the original version of the program is doing. Then, make sure you understand the constraints expressed above (i.e., that you understand the rules of this assignment). Then, work on the modified versions. Each version requires very little code (for me, each version required fewer than ten lines of new code). When I assign this program, I'll also give some additional hints.

When you are finished, email me your code to carl.sable@cooper.edu. Only email the three modified versions of the code and three screen captures showing the outputs of each version. Once I grade this, I won't allow re-submissions; so, if you want to talk to me about your code before submitting it, set up an appointment with me. Of course, you are also welcome to email me questions related to the assignment.