Kevin Martin

Syracuse University

CIS655 – Summer 2020, Tuesday @ 9:00pm EST

Homework 2

Question 2

For my IDE, I set it up using Microsoft Excel. As a quick background, I had worked in finance for about eight years before starting at Syracuse, and first learned basic programming concepts in visual basic (technically VBA, visual basic for applications). As such, I spent quite a bit of time in Excel throughout those years and thought it would be easy to make a nice looking and functional display. The result uses a combination of formulas and macros to achieve a user-friendly design.

The “language” supported here is a simplified version of MIPS. Each instruction is comprised of a 22 bit word: two bits for the type identification (I use 4 types as opposed to MIPS’s 3 types), five bits for the instructions, and then five more bits for (up to) three registers. Each Opcode is a unique string and identifiable on its own, and each register has its own unique string of five bits as well. Therefore, no two instructions will be the same, unless it is asking for literally the exact same thing. The “Manual” tab has the description and related Opcode identifiers for each of the 47 instructions supported. There are 16 available registers, also on the Manual tab. Also print statements need to have each word separated by a semicolon as opposed to a space, however register values can be printed at will. Instructions that begin with a “#” are ignored completely. Array indices are stored as an integer value in each register, and you access the array followed by an offset to get the exact element of the array desired.

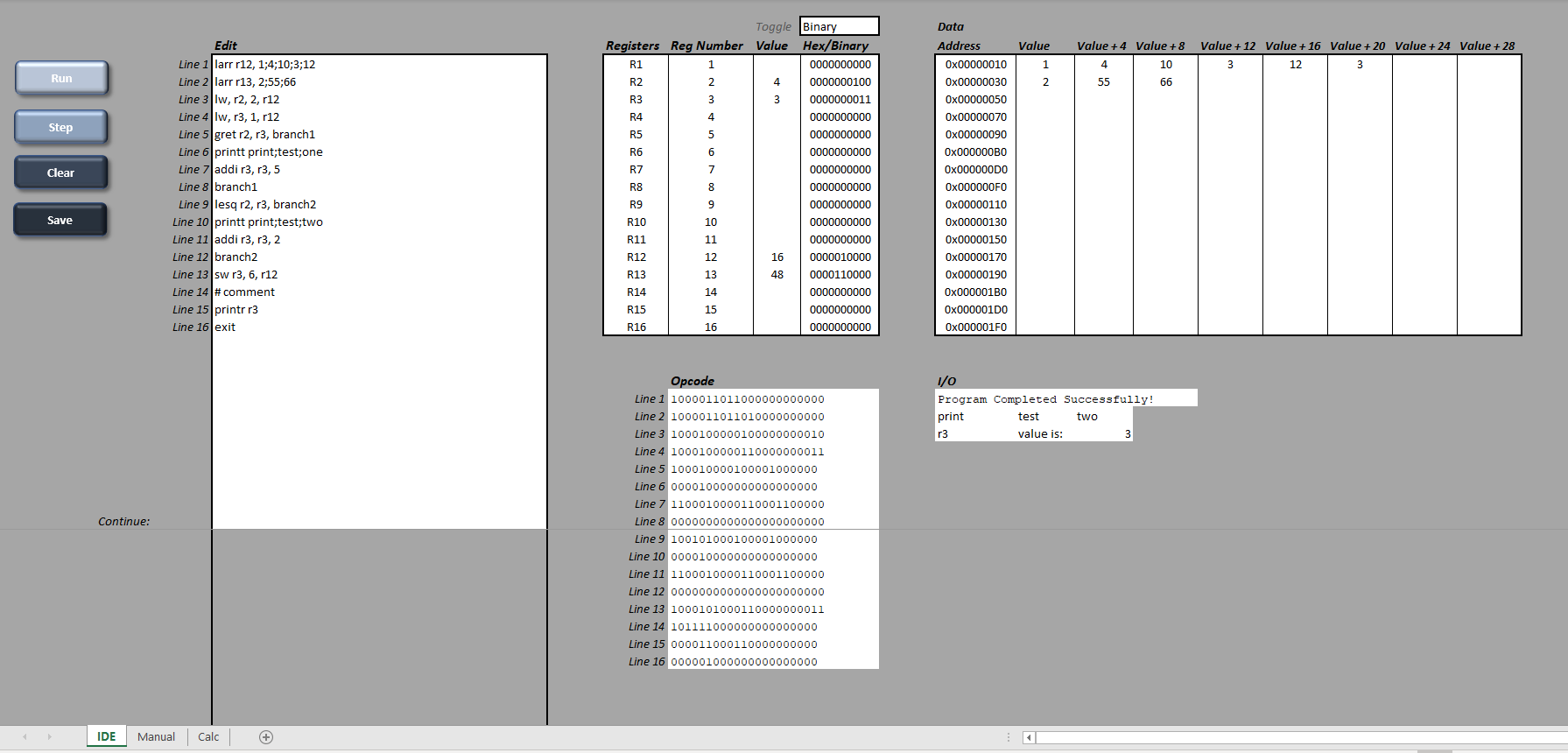
The “IDE” itself is on the IDE tab, and it also follows a similar outlay to MARS. The user only needs to use the whitespace provided in the “Edit” box, and the four buttons on the left-hand side of the screen. If the user needs more lines than provided by the whitespace, after the word “Continue” the gray space will automatically change to white to allow for more visually appealing typing. As far as the Data, this is simply comprised of 16 spaces in Memory, each with 7 four-bit values (identical to MARS here). Unfortunately, if the user needs more than those 16 spaces, memory is not available. Data is indexed using integer numbers during calculation, but is displayed using hexadecimal (again, following MARS).

The four boxes on to the right (Registers, Data, Opcode and I/O) all update automatically. To use, simply enter the commands, and click “Run”. The “Clear” button is helpful to see a fresh page, but it is not necessary to get an accurate line. “Save” simply saves the workbook according to regular Microsoft Office Save functionality (same as pressing Ctrl. + S on the keyboard). The Opcode section automatically translates each line in the edit window into its 22-bit word instruction. This section also changes from gray to white as needed. If the user chooses to enter any I/O (either print text or print register values), it will show up under the I/O section. Regardless of whether or not the user does use I/O, the phrase “Program Completed Successfully!” will be printed in that section as well.

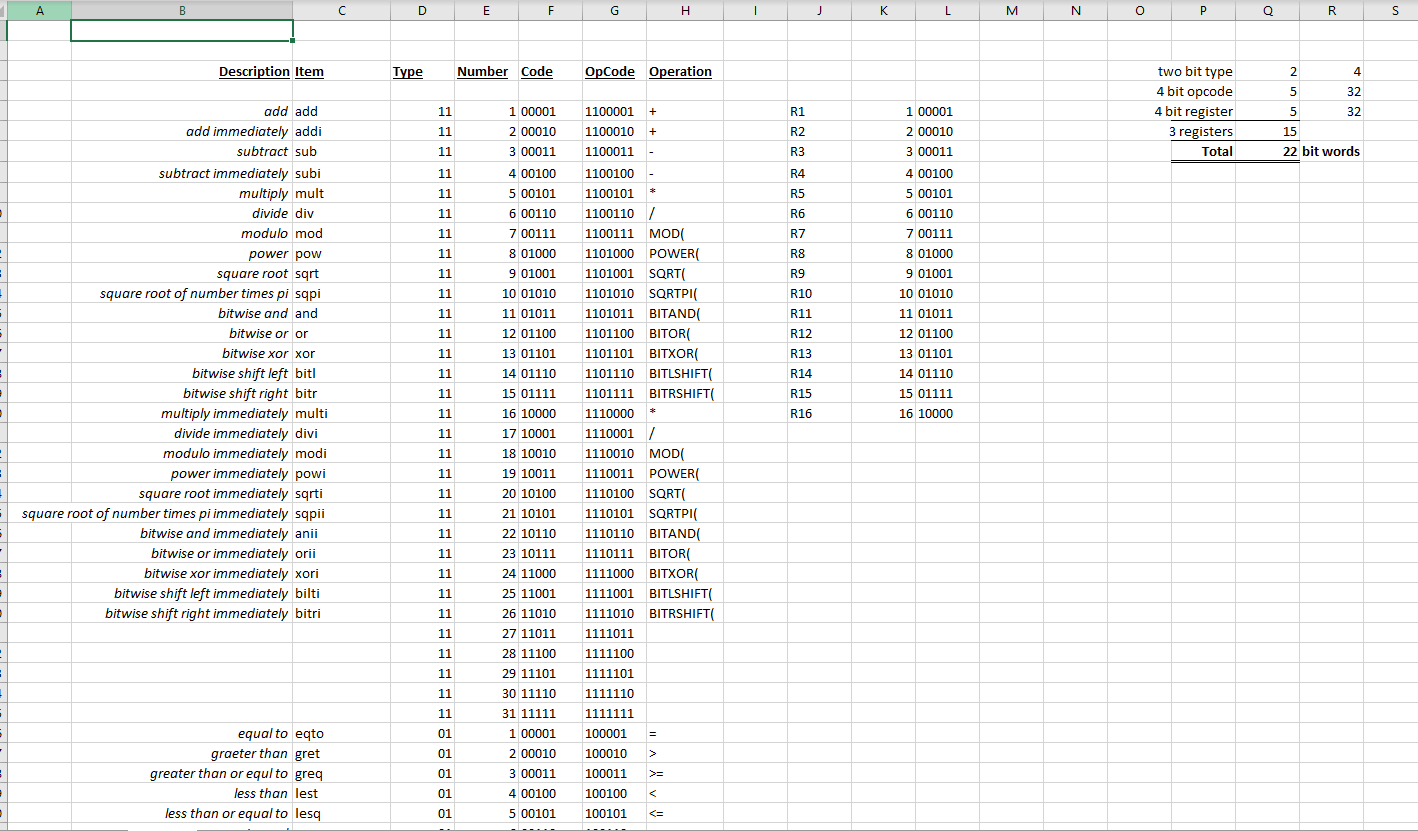
Finally, note the “Step” button. I am pleased with how this one works as it was sort of a nice to have feature I initially thought of. Step does the same thing as Run, except it pauses at each instruction to allow the user to observe the register and data updates. After the user presses “Next”, the next instruction is automatically updated.

For the screenshots, first the IDE tab where the user interacts with the program. Note that there are many formulas on this tab, the user does not need to be aware of any of them. Next, the Manual tab, with the 47 instructions available and their related codes. Finally, there is a hidden “Calc” tab which is used during program execution. This tab remains hidden, but I am including in the screenshots just to show the full path. Some of the formulas on the first tab were easier to display with this extra tab, hence why it is needed. All of the buttons are supported by macros written in VBA, so I will include all of those as well. The macros interact with the sheet and each other to form the complete IDE.

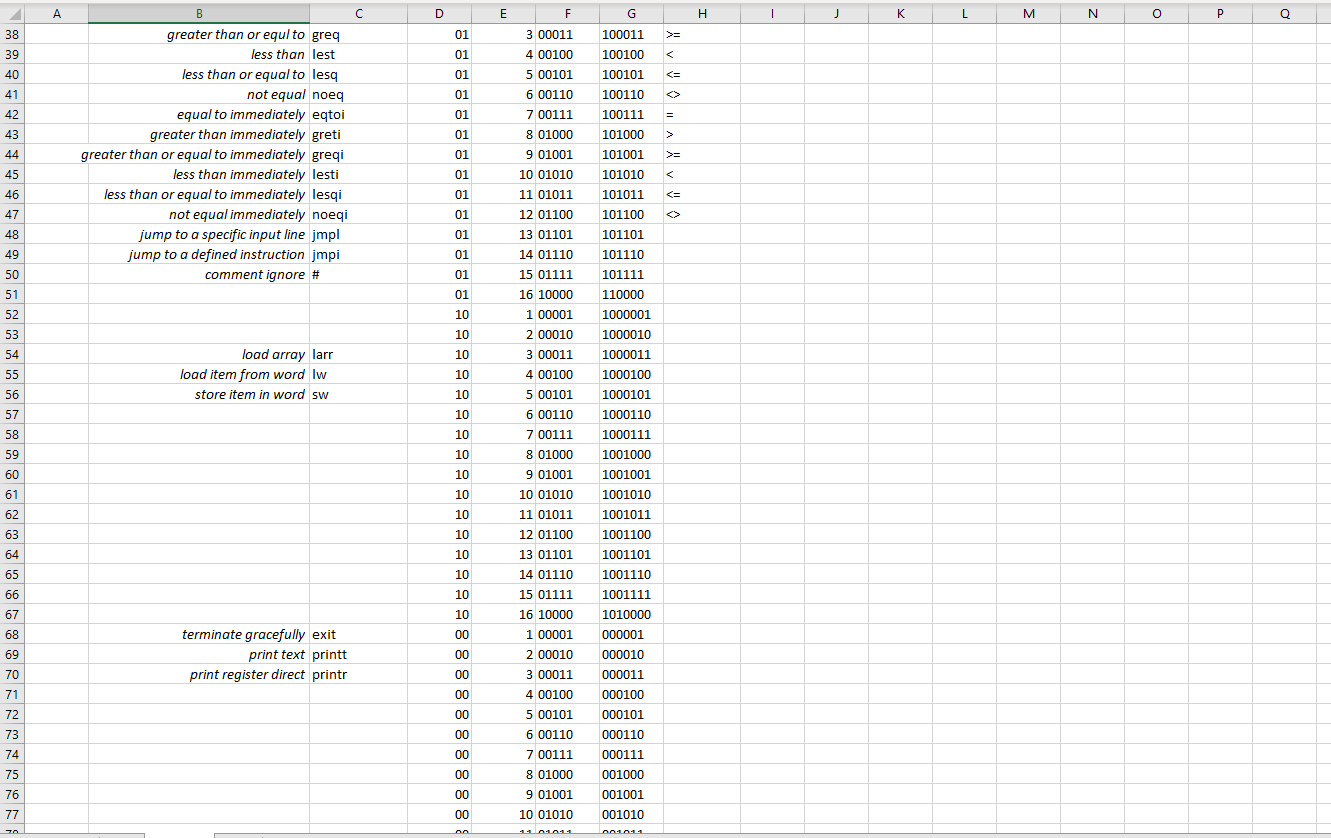
IDE tab:



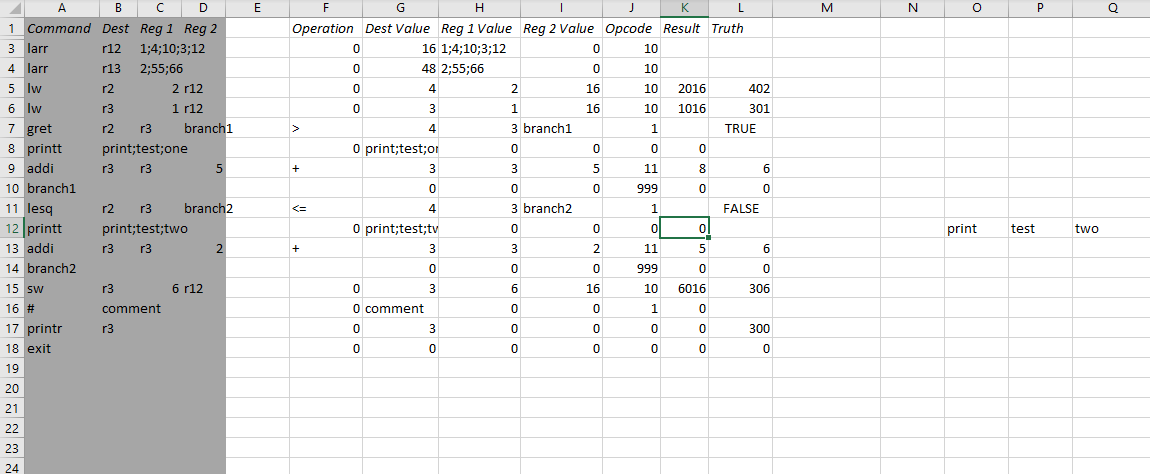
Manual page 1:



Manual page 2:

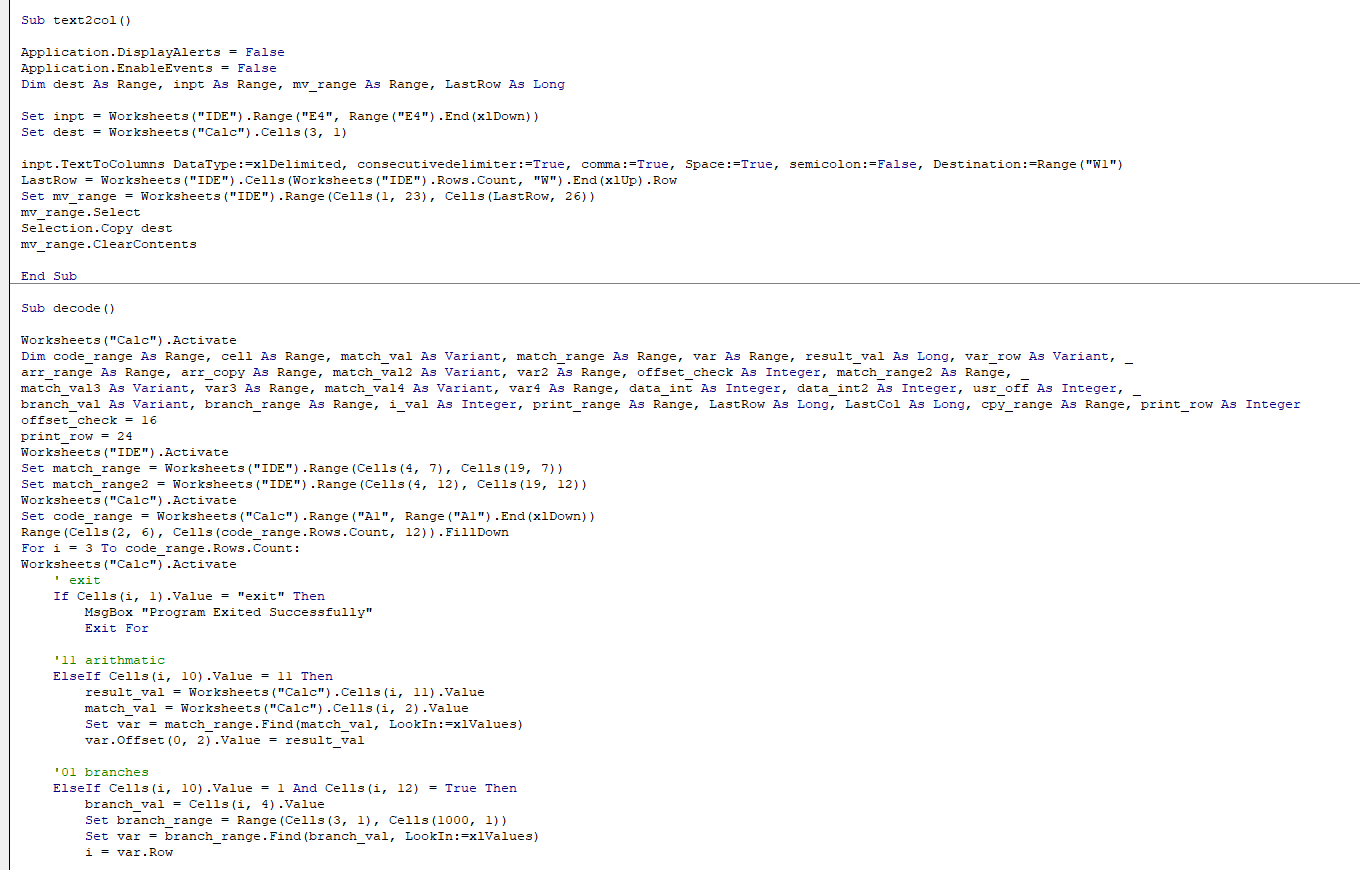


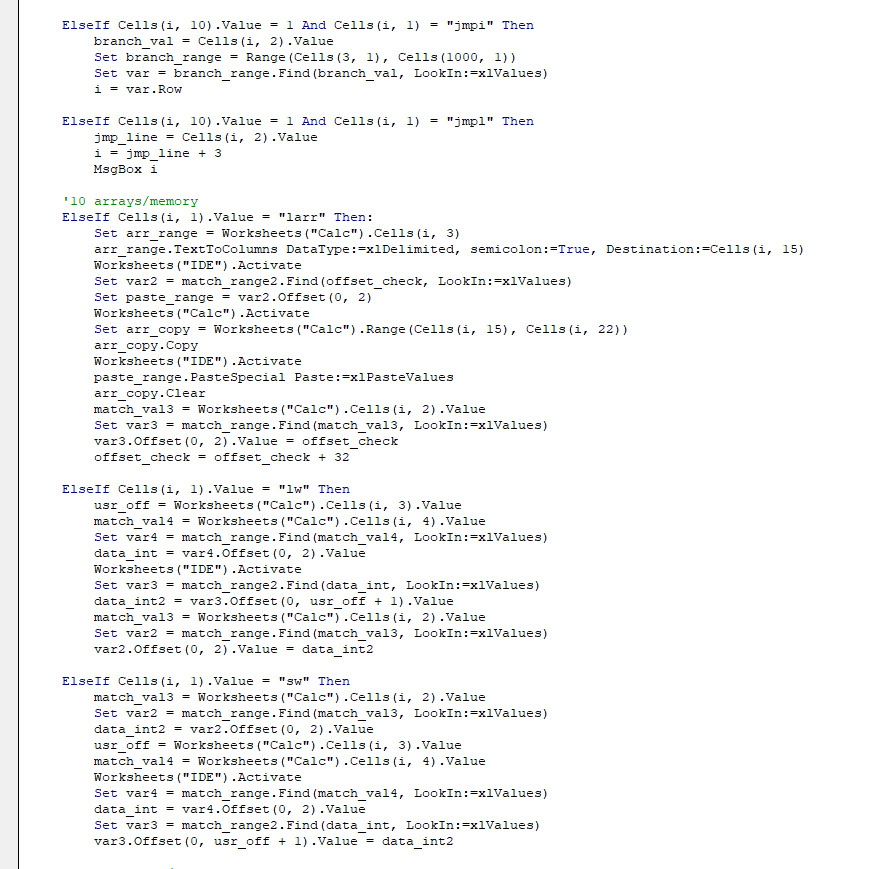
Calc tab (note: hidden at all times, only shown here for display purposes):

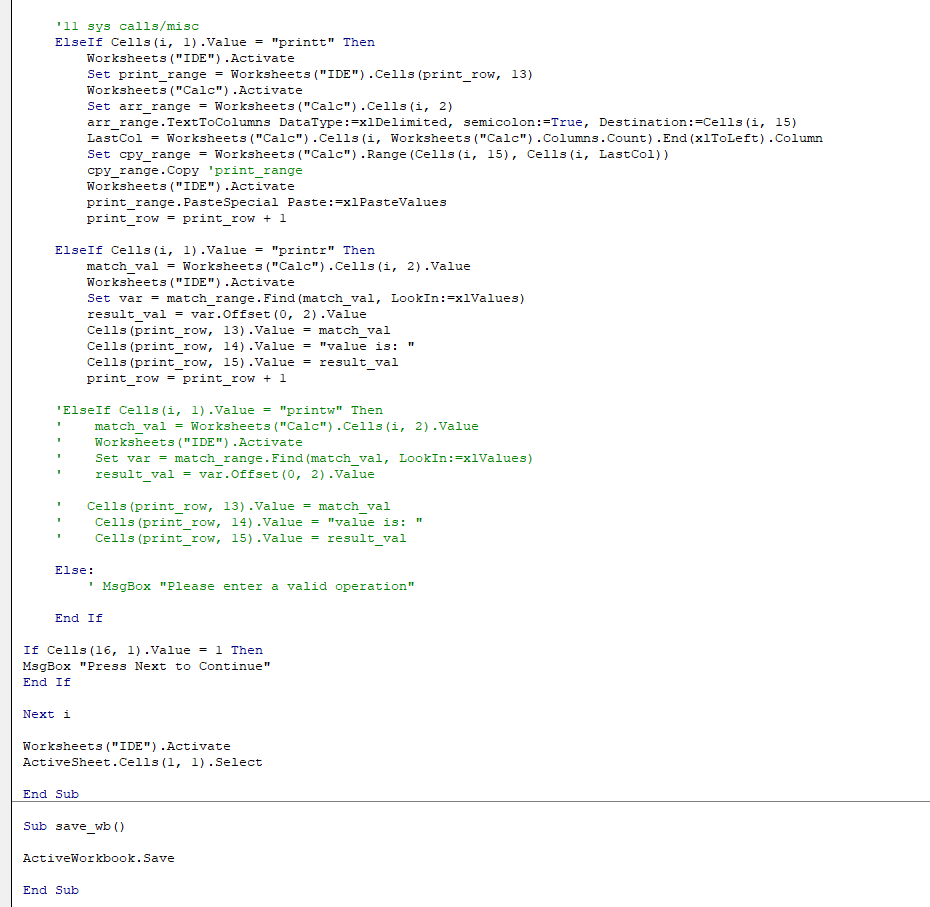


VBA/Macro buttons:









One small user defined function for parsing input text, as part of the macros:

