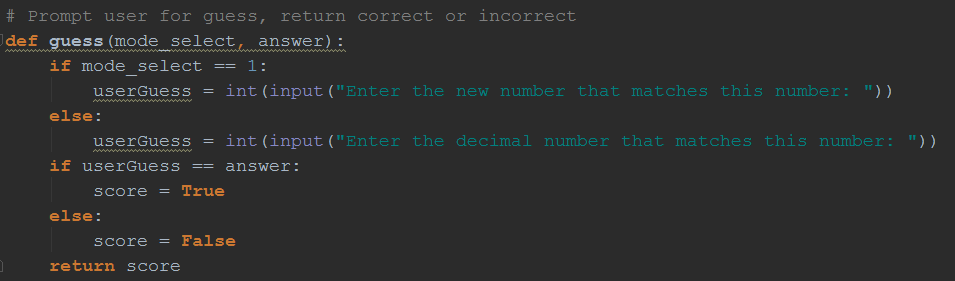
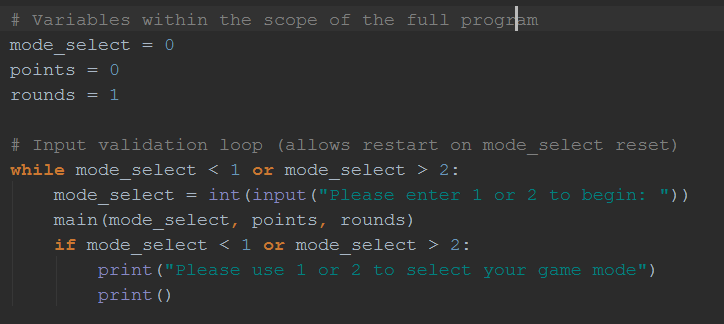
Brad Jones - CSC110 – 5/13/2017

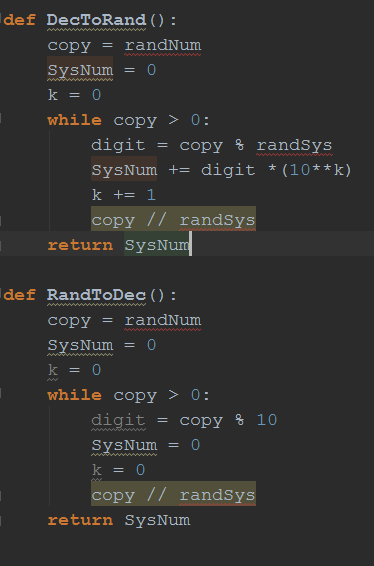
Integer base conversion game

I toughest part of this assignment for me was implementing the score/round keeping after I had already built most of the functions and structure. Things started to get hard to read after just added elements as I went along, so I rewrote the program from the beginning, adding a simple function with a Boolean return value to add score.



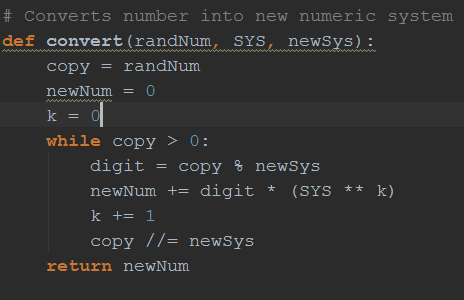
Then I had to work out the placement for the score/round accumulator, so that it did not reset in each loop. My solution was to call Main() from Menu(), and pass the parameters along. Menu() would not be called a second time.(and therefore, not resetting the score/rounds)



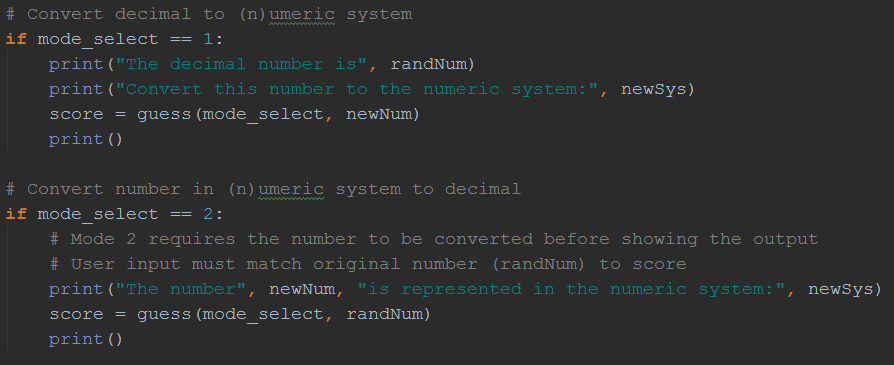


First attempt, not in final version

My first attempt used separate functions for mode1() and mode2(), but I had to figure out how to validate the input for mode2.



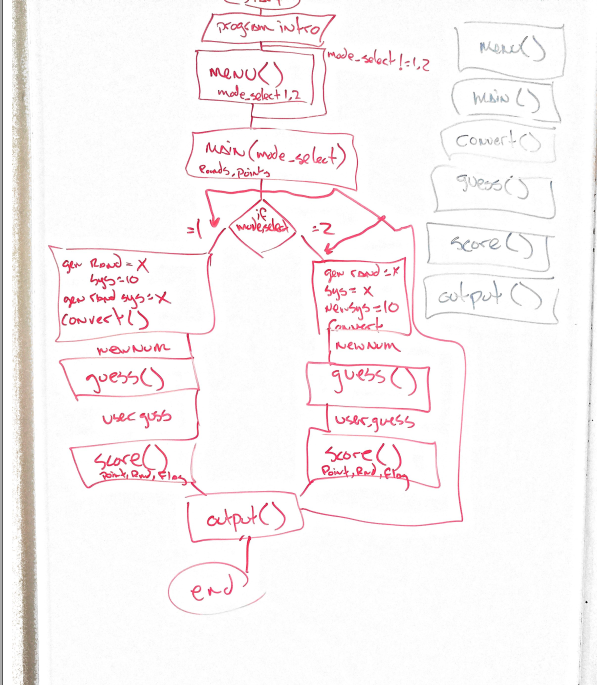
After stressing over that for a minute I decided a better approach would be to use only 1 function to convert the numbers, and have it accept parameters



Then I used if statements to separate the modes. Notice that in mode 1, randNum is displayed to the user to convert. In mode 2 however, the number is converted prior to showing the user.

|  |  |
| --- | --- |
| MODE 1 | MODE 2 |
| randNum is the number to be converted by the user | newNum is the number to be converted by the user |
| newSys is the base system to convert **TO** | newSys the the base system **FROM** |
| newNum is the correct answer | randNum is the correct answer |
| Score() accepts newNum to check if correct | Score() accepts randNum to check if correct |

Mode 2 will now never generate an invalid number.

This flowchart doesn’t look like much but it definitely helped.

Here is my final program:

