**INITIAL DESIGN PLAN**

* Import sys
* Determine if file is equal to 12 digits or not
* Calculate the modulo check number by
  + Adding odd positioned digits together and multiply by 3
  + Adding even positioned digits together
  + Adding the sum of the odd and even positioned digits together
  + Find the remainder when divided by 10
    - If remainder is 0, the modulo check number is 0
    - Otherwise, subtract the remainder from 10
  + Returns the modulo check number
* Split the digits into sets of 6 to have a left and right side
* Place them in a list and split the digits into a list of numbers
* Appropriately match the left digits and the right digits to their corresponding binary code
* Combine the binary codes from the left and right side to create a full binary code
* Import turtle
* Draw lines that match the binary code with the 0’s being white lines and 1’s being black lines
* Have the correct UPC code displayed with the drawn barcode
* Have a test suite
* Have a main function to call each of the functions
* Call the main function

**SUMMARY**

* The design will draw a barcode and display the UPC code if the file entered is valid. It will display an error message if the file has an invalid code. The design evolved immensely. I did not have the design take into account of invalid characters when determining if the code was valid or not. I only took into account whether they were 12 digits or not in the file. I also did not think about using dictionaries in my design plan like I did in my code. Overall, the outline of my design plan was needed, but much more detail was added in my code. I was able to do everything that was in my design plan, and display a barcode. I spent about 10-11 hours writing the initial design plan and the writing the code.

**IMPLEMENTATIONS**

* Accomplished checking to see if the code is 12 digits or not
* Calculating the modulo check number
* Drawing a barcode
* Displaying the valid UPC code
* Displaying an error message
* Added a small test suite to test my functions

**TESTING**

* 886971299922
* 071915024696

**FILES**

* upc-input1.txt
* upc-input2.txt

**ERRORS**

* The UPC code displayed does not match a normal barcode, as in the first and last numbers are not on the outside of the barcodes. They instead are in the middle with the rest of the numbers, and do not seem like a large enough font size.
* When exiting out of the turtle window, you have to restart the kernel if you want to re-run the program. Otherwise, it will make the program stop responding.

**COMMENTS**

* I thought the lab was fun to do, because I enjoy knowing that I was able to calculate the math and draw the corresponding barcode. It was a lot more difficult than I thought it would be to code this lab though. There was a lot that goes into it, but breaking it down step by step helped me a lot. The current TA’s and my former TA helped me a lot in finishing this program. I have had to use dictionaries in coding for work, but we have not learned it in class yet. We also have not learned about the pop method, and so that was something neat to learn and use in my code.