Overnight I ran levelMaker12 with 1-9 and 13 as inputs. 1-6 finished relatively quickly. 7 took almost 4 hours. I was able to use it a bit before accidentally messing it up (see 6.22.15\_Notes). 8 and 9 are yet to finish. 13 ran out of memory. It seems that time and space are key limiting components. Speeding up the truthValue finder or the gateCounter will help with time to an extent but not with space. If we can find a way to narrow what we build circuits from, this would reduce both time and space. 7-Gates built 228 circuits. Of the remaining 28, 10 could be found by inverting things found in the 7-Gates list. 7 more truths could also be found with 8 gates as confirmed by the Method B. So there would be at most 11 more truths remaining. Maybe if I look at the circuits not used in the dictionary with the truths, I will be able to find a shared characteristic that makes them not useful.

Bryan sent me the Json file with the gates used by the cello program. I will learn to open and read Json files and write a program that determines the number of gates used for each circuit to see if I am in fact finding the minimum circuits and to see in which cases the circuits are the same.

First I must find a jar file with all the necessary Json equipment. Bryan sent me the jar file I needed. I was successfully able to open the document and print out the number of gates needed for each truthValue set. Using the 228 circuits found from levelMaker12(7), I will see if the minimum circuit I found has the same number of gates as the one in the Json file. Since I just have a copy and paste of all the circuits, I will need to make a program that can read the file and convert it to a dictionary in java. Then I will count the number of gates and compare for each of the 228. Discovered one difference. In my version for 11100101, 7 gates are used while the cello version uses 8 gates.

Going to do a side mission. Goal is to find the minimum number of circuits needed such that by rearranging the inputs you would get all 256 circuits. This is solved by minNumberOfCircuitsFor256().

**The answer is 80 circuits.**

Was potentially able to find all circuits. It may not be the minimum. All the 8 circuits are the minimum. The 9s,10,11,12 may not be minimal although I highly suspect they are. Tomorrow I will compare it to the Json to see which were not the same number of gates.