Your submission will be an uncompressed Linux tar file, containing a single Logisim (.circ) file and a plain text file. If you submit the two files separately, instead of in a tar file, we will cheerfully ignore them. If you submit anything other than an uncompressed tar file, we will cheerfully ignore your submission. You can easily verify your file type via the Linux `file` command:

```
[wdm@centosvm DL01]$ file --mime-type DL01Solution.tar
DL01Solution.tar: application/x-tar
```

If you work in pairs, paste a copy of the Partner Template at the beginning of the text file, and fill in the names and PIDs of both students. Also, fill in your name(s) and PID(s) in the text fields in the supplied Logisim file. If you don't do this, only one of you will receive credit for the assignment.

The Logisim file will contain the solution to question 3, and must be named `2BitComparator.circ`. The text file will contain your answers to questions 1 and 2, and must be named `DL01.txt`. We don't care what you name the tar file. You can easily verify the contents of your tar file:

```
[wdm@centosvm C1]$ tar tf DL01Solution.tar
2BitComparator.circ
DL01.txt
```

Submit your file to the Curator system by the posted deadline for this assignment. No late submissions will be accepted. Submissions that do not conform to the stated requirements will be discarded.

You will submit your answers to the Curator System (www.cs.vt.edu/curator) under the heading DL01.

Designing a simple digital circuit

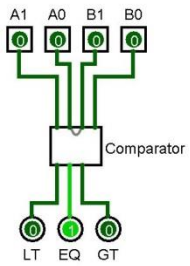
We all understand how unsigned integers are represented in memory (or we will review those things on our own.) We also know that if we have two unsigned integers, A and B, then either A < B or A = B or A > B.

So, we all understand the significance of the following partial comparison table:

A	B	LT	EQ	GT
00	00	0	1	0
00	01	1	0	0
.	.	.		
01	01	0	1	0
01	10	1	0	0
.	.	.		
11	10	0	0	1
11	11	0	1	0

Note how the rows of the table are ordered; the bits for A and B form the representations of the integers 0, 1, 2, etc. Put all your truth tables in that order, because it makes it much easier to check the results.

If we label the bits of `A` and `B` in the usual way (bit `k` corresponds to 2^k), we can envision implementing a circuit to compare two 2-bit inputs and indicate which of the possible relationships holds, with an interface like this:



1. [16 points] Create a complete truth table that defines the relationship between the inputs and outputs for the 2-bit comparator shown above. Use the same labels as shown in the circuit interface diagram. Be very careful with this, since the remaining parts of this assignment depend on the correctness of your truth table.

2. [40 points] From the truth table, derive a sum-of-products Boolean expression for each of the outputs, `LT`, `EQ` and `GT`. Since you have to write this in a plain text file, use `'~'` for NOT, `'*'` for AND, and `'+'` for OR.

Use the axioms and theorems of Boolean Algebra to simplify each of these Boolean expressions as much as possible (but keep them in sum-of-products form).

3. [44 points] Use Logisim to create an implementation of a 2-bit comparator, based on the simplified Boolean expressions you derived above. You must conform to the interface shown above. We will grade your solution by generating a truth table from it (see below) and checking the truth table with a program. So, you must have the columns and rows in the order shown in question 1.

You are restricted to the components available from the Wiring, Gates and Base Libraries in Logisim; you may not use any Logisim component that performs an arithmetic calculation or comparison (as opposed to a logical calculation).

Hints:

If you consider the available logic gates, you may find a simpler solution for some of the outputs than is given by your answers to question 2. If you do, it's OK to base your circuit for those output on those gates instead of your answer to question 2.

You can use Analyze Circuit from the Project menu in Logisim to generate a truth table for your circuit. If you do all this correctly, that should exactly match your truth table from question 1 (which is good if and only if your truth table is correct).