1. Please enhance your logic simulation program to support the following features:
   (a) Make the input pins (and the links connecting to these pins) of each gates or composite gates graphically distinguishable. For example a three-input AND gate should be drawn with 3 different input pins. [3 points]
   (b) Support two distinct states that allow editing of circuits in different modes: (1) layout state, and (2) connection state. Running simulation of logic circuits should work for both states. You should provide GUI (e.g., toolbar buttons) to enable the switching between these two states. The detailed operations allowed in each state are defined in the following questions. Please apply State pattern in your design. [3 points]
   (c) The basic operations of layout state are the same as the operations described in the last homework. That is a device (or a number of devices) should be able to be selected by the left button of the mouse, and a selected device (or a set of selected devices) can be moved (dragged and dropped) from one position to another also by the left button of the mouse. [1 points]
   (d) In connection state, selecting devices is not allowed. However, your program should allow users to connect two devices by linking an input pin and an output pin. Please use the left button of the mouse to select the target input pin and the target output pin to be connected (note: using dialogs to identify pins are not allowed). The connection of two input pins or two output pins should not be allowed. [3 points]
   (e) Add GUI (e.g., buttons) to support the ability of adding new gates (AND, OR, NOT, XOR, NAND, NOR, HA, and FA) to the logic circuit. Adding new gates are allowed in both layout and connection state. The default number of input pins for AND gate and OR gate are two. [3 points]
   (f) In layout state, add a GUI (e.g., a button) to enable a selected OR gate or AND gate to increase the number of its input pins. You are required to use RTTI of C++ to identify whether a selected gate is indeed an OR gate or an AND gate. [3 points]
   (g) In layout state, add a GUI to group a number of selected devices into a single device. A grouped device should behave the same as a regular device. That is a grouped device should be capable of being moved, linked, and grouped recursively by the operations described above. A grouped device should be implemented as an instance of a composite object. [3 points]
   (h) In layout state, add a GUI to enable copying of a selected device (including a grouped device described above). Please apply a Prototype pattern in your design. [3 points]
   (i) Support the ability of deleting a selected device (and the links associated with this device) in layout state, and the ability of deleting a selected link in connection state. [3 points]

2. Add a C++ namespace called LogicSimulation and put your simulation programs in this namespace. Except for few GUI related programs, almost all of your programs
should stay in the LogicSimulation namespace. [3 points]

3. You should make your logic simulation program described above as robust as possible. That is it should not easily get crashed and there are no obvious bugs or memory leaks. [4 points]

4. Please keep your unit tests running. You do not need to add any new tests for the new programs written in this homework. However, you should maintain your unit test programs written for homework #4. [2 points]

5. Draw the class diagram of your implementation. You do not need to draw the class diagram for unit tests. You should simplify the drawing of your class diagram by ignoring unimportant member variables or functions. [3 points]

6. Bonus: when the simulation of propagation is activated, show the path with the largest propagation delay graphically (e.g., show gates and links with a different color). [3 points]