Basics of LabVIEW Environment

**1. Logic Functions**

The user selects one of the three logical functions and exposes its arguments. The program performs calculations and outputs the result to a Boolean indicator, which should change its color (green / red) and text ("TRUE" or "FALSE") depending on the state. Function list:

F=







**2. Arithmetic&Logic Calculator**

The user enters two numbers in binary (numbers, not boolean), selects one of the operations: AND, OR, XOR, NAND, NOR, addition, subtraction, multiplication, and then clicks the "Calculate" button. The result is displayed on a numeric indicator in binary form.

2a \*. Make the result output window simultaneously be used as one of the operands (like in a real calculator).

**3. Fibonacci Numbers**

The user enters the number of a number in the Fibonacci sequence (1, 1, 2, 3, 5, 8 ...), and the program displays all numbers up to the one indicated on the graph (XY graph). Is your program calculating F (100) correctly?

**4. Graph of Trigonometric Functions(Formula Node)**

The program displays the graph of the specified function on the chart (Waveform Chart). The user can select any function and its parameters: amplitude, frequency and phase. List of required functions:   
f (t) = Asin (ωt + φ); Acos (ωt + φ); Atg (ωt + φ); Actg (ωt + φ); Asin (ω1t + φ1) + Bcos (ω2t + φ2); Atg (ω1t + φ1) + Bctg (ω2t + φ2). It is enough to display two or three periods of the function on the graph.

**5. Lissajous Figures(XY Graph)**

Lissajous figures, as you remember from the physics course, are given by a system of parametric equations: x = sin (a1t + b) y = cos (a2t)

The user enters the coefficients a1, a2 and the phase shift b (in degrees), the program draws the resulting Lissajous curve on the XY graph. It’s better to implement phase shift control in the form of a "knob" (Knob).

**6. Lower&Upper Case**

The user enters a string of characters, the program reads it and replaces the lowercase letters in it with uppercase, and uppercase with lowercase, and displays what happened on the indicator.

**7. Running Line**

The user enters a character string of any length, and the program displays it as a "running line" to the String indicator. Line speed must be specified (in characters per second).

**8. State of Plane(Cluster and Formula Node)**

Create a cluster - type definition (Typedef), to display the state of a plane: current coordinates, velocity, departure and arrival points, number of passengers and crew on board. The plane flies from New York to Los Angeles, simulate the changing of states during the flight (in accelerated time, of course).

**9. Move or Create File**

The user lists all files and folders in a desired location. In addition, to create a file in this location user should enter the file name or to move an existing file user should just enter the file name and path to be moved to.

**10. Alarm clocks (Global Variable)**

The user should select an alarm clock from a listed alarm clocks by time(listing can be done with read and write to spreadsheet functions). Then via a global variable, this time value should be transferred another VI where the actual alarm clock function will be implemented.

**11. Kettle (subVI and CSV file)**

The program should include two VIs amster and its subVI. SubVI should simulate a kettle operation and master should write the operation data(kettle condition-On/Off, kettle temperature, outside temperature, real time) to a CSV file by time(for example by 100ms intervals).

The subVI should operate as follows: The user should select an outside temperature and kettle temperature should be equivalent to this value at the beginning. When start buton is pressed, temperature should gradually increase, if kettle temperature reaches until 100 °C as a maximum value then program should halt automatically. When kettle is at “Off” condition, if it’s hotter than outside temperature it should gradually cool down until it decreases to outside temperature. Note that if you will create a subVI in a loop(while,for) in master VI, this loop affects all subVI elements as they are located inside this loop in subVI itself.

State Machines

**12. The Red LED**

The user must push three buttons in expected order (like a password). After that a red LED indicator (boolean) lights up (in “CLOSED” condition) then a by turning on a switch, condition of the boolean should be changed to "OPEN". If the buttons are pushed in the wrong order, the indicator should stay disabled (disabled and grayed, this is one of the properties of property node).

**13. Crosswalk**

There are two traffic lights at the pedestrian crossing: for cars and for people. At each traffic light, red is on for 5 seconds, yellow - 2 second, green - 5 seconds. Simulate this situation. Add a button at pedestrian traffic lights to turn the lights to green when they are green for vehicles and red for pedestrians.

**14. Vending Machine**

The old vending machine has two options - soda with syrup (3 cents) and soda without syrup (1 cents). If someone puts 1 cent into the slot, then the sequence is as follows:

- water is poured into the internal working volume (0.5 seconds),

- carbonated from the siphon (0.5 seconds) and

- poured into a glass (1 second)

- after that the "ready" indicator lights up and stays on for 3 seconds.

If a person drops 3 cents, then syrup is added to the glass beforehand (0.5 seconds), and then as in the previous case. In order to see how the sequence of actions occurs, display the current state on an indicator of type String.

20a \* Add a money meter and sensors for the flow of water, gas from a spray can and syrup for the machine.