Assignment1

1.
(1). Build a subvi named thermometer.vi, which randomly output a Celsius temperature value in between the lower and upper bound set with input parameters of lower bound value, and upper bound value (2 inputs and 1 output). Please set the two inputs to be "required"
(2). Build a subvi named toFahrenheit.vi, which converts a Celsius temperature reading to a Fahrenheit temperature reading. (one input and one output). Please set the only input to be "required"
(3). Build a top level vi, continuous shows the temperature values got from thermometer.vi with a waveform chart in a while loop until the user press stop button. On the panel, user can select the display of temperature scale in Celsius or Fahrenheit and summer or winter mode with different upper/lower bounds preset inside the program.
(4). Show second waveform chart for the current temperature, the running average temperatures in past 5 loops, the average values, the maximum and minimum values of all past temperature values on the front panel and on the waveform chart with multiple-plot. Please do not use the built-in mean and max-min function node and implement the minimum, maximum function by yourself (with shift register and comparison in the loop).

2. Write a dice rolling game with two players which stops when user pushes the stop button. Time delay can be used to slow the game down.
This game should include:
(1). A sub VI which randomly rolls a dice for each player, when player1 or player2 push the roll buttons on top VI.
(2). A second sub VI which compares the result to mark the winner, or if it is a tie game.
(3). On the top VI, record the average dice number rolled for each player. Calculate the win rate and tie rate. Also record the past 5 dice numbers for each player.
(4). Show the win rate of player 1 and player 2, and the tie rate with a multiple plot waveform chart.

Assignment2

1. Write a VI to simulate the pedestrian traffic light in Taipei city which has an count down timer display showing two digits remaining time, and an 16x16 LED panel display showing an animation of a little green guy (小綠人).
(**hint: Use a 3d boolean array constant to store the animation data.**)
The VI has a numerical control to set the final count down period. In the beginning of count down, the LED panel shows animation of the little green guy walking. When the remaining time is less then 10 seconds, the LED panel shows animation of the little green guy running.

2. Bubble sorting compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed
For sorting the array 51428.
First Pass
(5 1 4 2 8)->(1 5 4 2 8), 5>1, swap
(1 5 4 2 8)->(1 4 5 2 8), 5>4, swap
(1 4 5 2 8)->(1 4 2 5 8), 5>2, swap
(1 4 2 5 8)->(1 4 2 5 8), 5<8, no swap
Second Pass
(1 4 2 5 8)->(1 4 2 5 8), 1<4, no swap
(1 4 2 5 8)->(1 2 4 5 8), 4>2, swap
(1 2 4 5 8)->(1 2 4 5 8), 4<5, no swap
Third Pass
(1 4 2 5 8)->(1 4 2 5 8), 1<4, no swap
(1 2 4 5 8)->(1 2 4 5 8), 2>4, no swap
Please Generate 1024 0~10000 random values and put them into a 1D array.
Visualize the sorting process on a waveform bar graph with a slightly delay.

3.
(1) Write a sub-vi named “score generator”, which can generate a random score number between the minimum and maximum score values given from the inputs. Set the two inputs as "recommended", and set the default value to be 20 and 80 if user does not wire values.
(2) Write a sub-vi named “score sorter” based on the bubble sorting VI you

wrote in 2., which can sort a 1D cluster array. The cluster has two elements. The first element is score and the second element is student ID number (1~50).

(3) Write a top-level vi, which call the score generator to generate 50 scores and then sort the scores with “score sorter”. Display the sorted result as a table with 3 columns, index number, student ID number, and score. Draw a top-10 scores bar graph, and removed duplicated scores.

4. A child is climbing a stair with n-steps. He can climb in 1-steps, 2-steps or 3-steps at a time. Find all combinations of he can climb the stair. Ex: For a 5 step stair, he can climb in {1,1,1,1,1}, {1,1,1,2}, {1,2,2},{2,3} (no permutations, treat {2,3} and {3,2} as the same solution) Do exhaustive searching, and show solutions in a 2D numeric array like,

1 1 1 1 1
1 1 1 2 2
1 2 2 2 2
1 1 3 3 3
2 2 3 3 3
{2 2} for a 2 steps climbing and {3 3 3} for a 3 steps climbing

5. N Queen problem can be describe as: Place N queens on an (N by N) chess board such that none of the queens attacks any of the others. Find all possible solutions of 8 queens on an 8x8 chess board. Draw all solutions on a 3-dimension LED boo lean array indicator.

Reference: http://www.aiai.ed.ac.uk/~gwickler/eightqueens.html

Assignment3

1. Insertion sorting is one by one remove the element from the original array and insert it into the proper location which is larger than the previous value and smaller than next value in the array.
For sorting the array 52314.
2 is inserted before 5, resulting in 25314
3 is inserted between 2 and 5, resulting in 23514
1 is inserted at the start, resulting in 12354
4 is inserted between 3 and 5, 12345.
End of sorting.
Generate 1024 0~10000 random values and put into a 1D array. Visualize the insertion sorting process with waveform graph and slightly delay.

2. Selection sorting is making only one exchange for every list pass through.
On every pass through, the selection sort searches for the largest number and places it at first location by swapping.
On the second pass through, the second largest is found and placed in its location after the largest by swapping... and so on...
The final item would be placed in (n-1) steps.
Generate 1024 0~10000 random values and put them into a 1D array.
Visualize the sorting process on a waveform graph with slightly delay.
First Pass
(68,2,93,45) -> (68,2,45,93), 93<->45 swap
Second Pass
(68,2,45,93) -> (45,2,68,93), 68<->45 swap
Final Pass
(45,2,68,93) -> (2,45,68,93), 2<->45 swap

3. Write a **BenchmarkSorting.vi** to benchmark the performance of bubble (you have done this in HW2), selection, and insertion sorting.
The vi will generate 1024 0~10000 random values first and feed the same dataset to the three sorting algorithm.
Package the three sorting algorithm into subVIs (named bubble sorting, selection sorting and insertion sorting) and run them one by one. Show the time required to finish sorting with the 3 different sorting methods individually.
(Note1: please remove all wait function inside the loops when sorting)

(Note2: you need to hand in four VIs in question 3)

4. Write a count down timer VI with 3 numerical display controls in hh:mm:ss format to let user enter the time to counting down from. The VI has a Boolean button start/stop controlling start or stop (pause) count down timer.
Use local variables (do not use shift registers) to read the 3 numerical controls and update the count down time per second. Beeps for 3 seconds by using beep.vi when counting down time reach 00:00:00. The time value then count up and showing a negative sign with a rectangular LED in front of hh:mm:ss.

5. Rewrite the reaction timer VI in Lab 1 with sequence for timing and local variable for the LED control.

Assignment4

1.

Design a VI to predict the trajectory of a cannonball shooting out of a canon on the ground. Draw a horizontal ground surface in the front panel with decoration elements. With four numerical controls, the initial velocity, the initial angle, percentage of energy loss when the ball hit the ground, and the percentage threshold of energy remained for the cannonball not bouncing again, try to draw the trajectory of the cannonball with a LED in the front panel and update the coordinate with property nodes. Please consider gravity but ignore the air drag effect.

Try to tune the parameters to have the ball bounce 3 times before it stay on the ground, and make the final parameters as default values saved with the VI.

2.

```
Apr 26 21:31:34 filterlog:
129,16777216,,1425041642,hn0,match,pass,in,4,0x0,,128,16110,0,none,17,udp,36,10.2.10.175,255.255.255.255,60221,3956,16
Apr 26 21:31:34 filterlog:
129,16777216,,1425041642,hn0,match,pass,in,4,0x0,,128,4117,0,none,17,udp,78,10.2.10.1,208.93.4.206,137,137,58
Apr 26 21:31:32 filterlog:
130,16777216,,1425041643,hn0,match,pass,in,4,0x0,,128,18168,0,DF,17,udp,351,10.2.10.10,10.2.23.123,53,59118,331
Apr 26 21:31:32 filterlog:
129,16777216,,1425041642,hn0,match,pass,in,4,0x0,,128,10627,0,none,17,udp,82,10.2.10.10,140.112.2.2,59087,53,62
Apr 26 21:31:32 filterlog:
129,16777216,,1425041642,hn0,match,pass,in,4,0x0,,128,10626,0,none,17,udp,98,10.2.10.10,140.112.2.2,60847,53,78
Apr 26 21:38:10 filterlog:
105,16777216,,1425041627,hn1,match,block,in,4,0x2,0,109,9812,0,DF,6,tcp,48,207.46.13.171,10.2.10.15,25309,80,0,SEC,27892632,,8192,,mss;nop;nop;sackOK
Apr 26 21:38:08 filterlog:
129,16777216,,1425041642,hn0,match,pass,in,4,0x0,,128,23124,0,DF,6,tcp,52,10.2.10.172,207.46.153.155,60590,443,0,S,2141064781,,8192,,mss;nop;wscale;nop;nop;sackOK
```

```
Apr 26 21:38:08 filterlog: 130,16777216,,1425041643,hn0,match,block,in,4,0x2,0,128,8596,0,DF,6,tcp,48,10.2.10.2,10.2.102.10,41875,88,0,SEC,3256166066,,8192,,mss;nop;nop;sackOK
Apr 26 21:38:07 filterlog: 130,16777216,,1425041643,hn0,match,pass,in,4,0x2,0,128,8592,0,DF,6,tcp,48,10.2.10.2,10.2.102.10,41874,88,0,SEC,2184502132,,8192,,mss;nop;nop;sackOK
```

The above text are derived from a section of firewall logs. Each line of the text above logged the important information of a packet being granted or blocked passing through the firewall.
The fields of interest are explained with the example

```
Apr 26 21:38:10 filterlog: 105,16777216,,1425041627,hn1,match,block,in,4,0x2,0,109,9812,0,DF,6,tcp,48,207.46.13.171,10.2.10.15,25309,80,0,SEC,27892632,,8192,,mss;nop;nop;sackOK
```

Apr 26 21:38:10 : The date and time for the packet to be logged.
hn1 : The interface of the firewall being logged. The field is always in the form of hnx, and x denotes the x-th network interface of the firewall.
block : The packet is being rejected passing through the firewall. The other possible value of the field is pass, denoting the packet is being granted passing through the firewall.
tcp : The internet protocol. Possible values includes tcp and udp for this field.
48 : The length of the packet
207.46.13.171: The source ip
10.2.10.15: The target ip
25309 : The source port.
80 : The target port.

Please ignored all the other field unmentioned.
Please copy and paste the above firewall logs into a string control in Labview. Separate a line with \n (or \n\r) and process a line in each loop with a while loop. End the while loop when the line retrieved is empty.
Use regex to match above mentioned fields of interest in each loop (line).
Output all the field of interest to a table (a 2D array of string indicator) to

show the above logged information in the form of
Date-time Interface-name Protocol Source-ip Source-port Target-ip Target-port Length Pass-or-block and save the result to a text file.

Assignment5

1. Please finish the on class example with state machine style VI to process a Boolean button cluster with “login”, “configure data acquisition”, and “data acquisition” buttons. Modify the VI to have the following extra functions.
(1) A login popup window to request user enter username and password when the login button is pressed. Hardcode the username and password in the login subVI and send back information when correct login username and password is entered.
(2) If the configure data acquisition or data acquisition buttons are pressed and the user is not previous login-ed, show the login popup window to request the user login first. If the user failed login, the user login window will popup again. If the user login successful, the program will continue the function corresponding to the button user pressed.
(3) If the "configure data acquisition" button is pressed, show a centered window with two numerical inputs for sampling rate (samples/sec) and samples per acquisition, with default values of 1k samples/sec and 10000 samples per acquisition. The window dismissed after "OK" button is pressed.
(4) If the "data acquisition" button is pressed, show a centered window with a waveform graph shows a sinusoidal waveform with the samples (horizontal axis) with the "samples per acquisition" parameter user set in "configure data acquisition". Convert the unit of horizontal axis to seconds with the "sampling rate" parameter user set in "configure data acquisition". The window dismissed after "OK" button is pressed
(5) After login, if user press "configure data acquisition", proceed to "data acquisition" step after the configuration window is dismissed.

2. Write a **calculator** as the **Windows Calculator** program with basic “C +-*/.= backspace“ function keys and 0~9 numeric keys. The calculator continuous runs until a “off” button was pressed.
Please write the program using **“state machine”** architecture. (a separate VI).
The +-*/.= (map to enter key) and 0~9 keys can also be enter with physical keyboard