while Loops and for Loops in LabView

Looping structures are patterned after C or C++

while (*Boolean condition*) statement;

while (i < 20) {

A compound statement is a bunch of statements enclosed by curly braces!

}

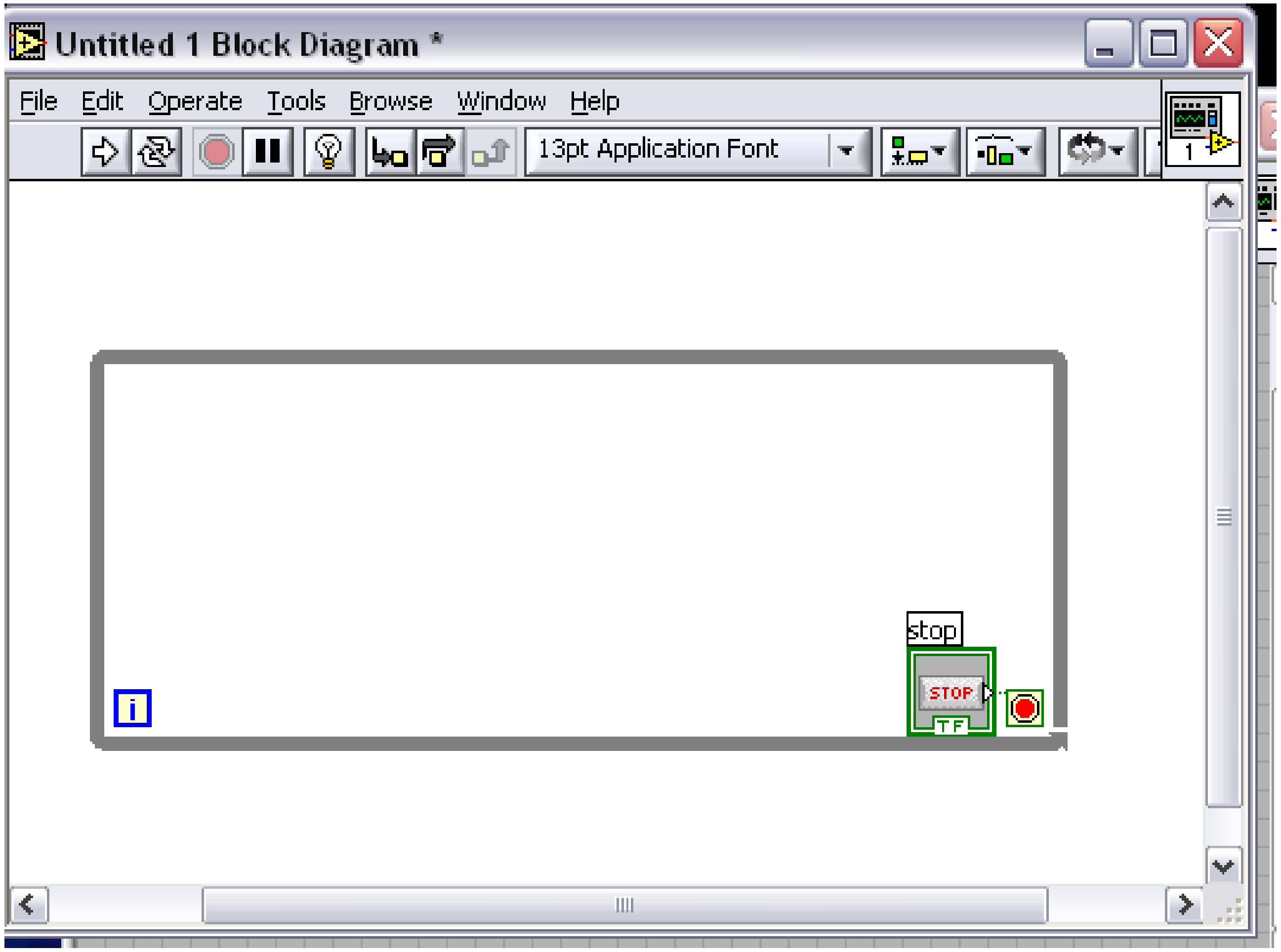
- A Boolean condition is either true or false.
- The program stays in the loop so long as the Boolean condition is true (1).
- The program falls out of the loop as soon as the Boolean condition is false (0).

```
int i                                  ;
i = 0                                  ;
while (i < 10) {
++i                                     ;
printf (“i is now = %d \n”,i)   ;
}
```

would produce:

```
i is now 1
i is now 2
i is now 3
*
i is now 10
```

LabView while Loop


Untitled 1 Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
stop
STOP
TF
i

Repeat whatever VI's are in the box until the Boolean function (including STOP, and maybe depending on i) is true/false.

Infinite while loop

```
while (1) {



}
```

This code repeats whatever is between the curly braces until $1 = 0$. (i. e. Until you unplug your computer.)

This is the most commonly used while loop, by far!

for (initial statement;*Boolean condition;*iteration statement) body statement;

```
for (i = 0; i < N; ++i) {

        A compound statement is a bunch of
        statements enclosed by curly braces!

}
```

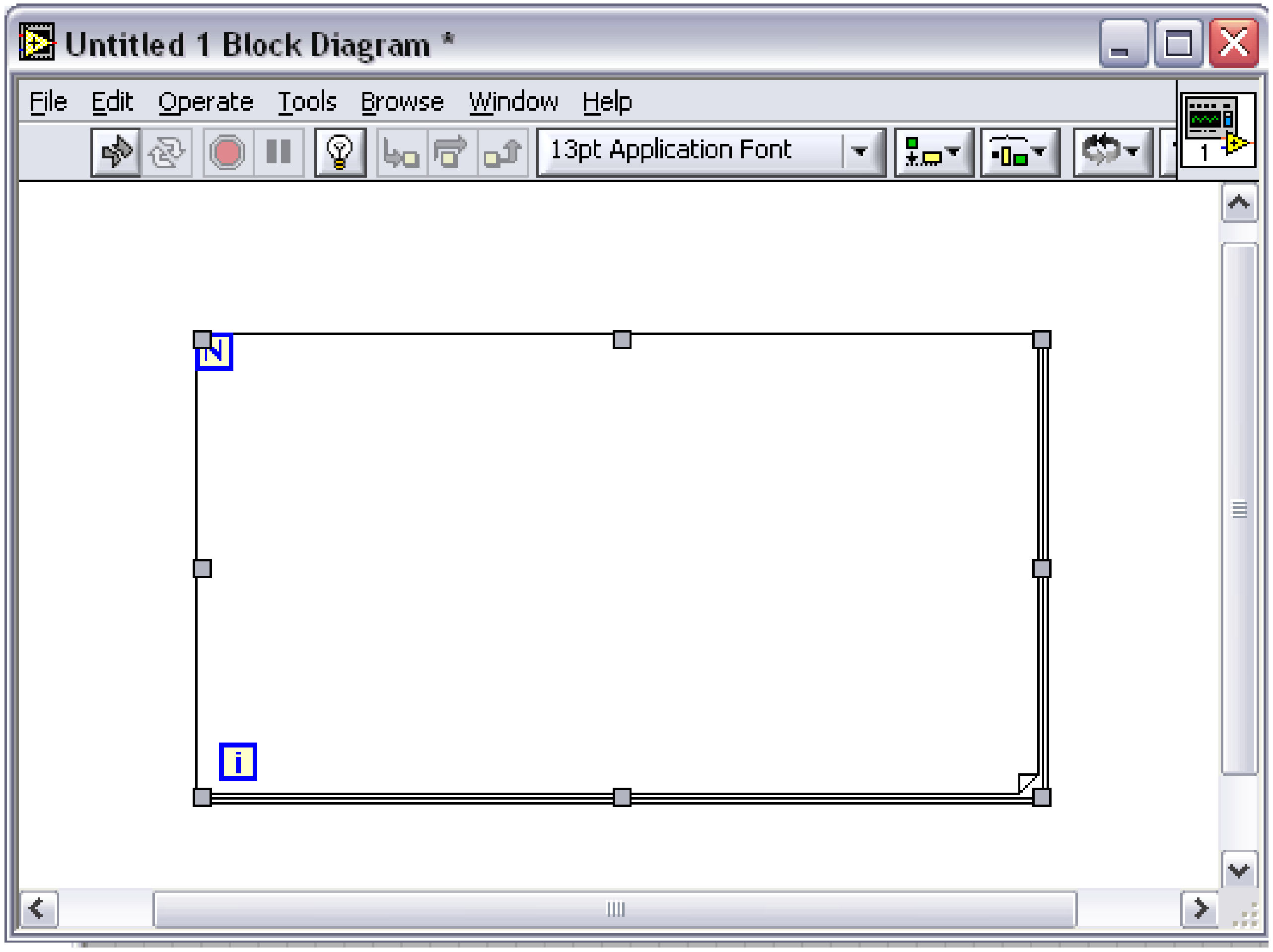
- A Boolean condition is either true or false.
- The program stays in the loop so long as the Boolean condition is true (1).
- The program falls out of the loop as soon as the Boolean condition is false (0).

```
int i , N                                        ;
N = 11                                      ;
for  (i  =  1; i < N; ++i) {
printf (“i is now = %d \n”,i)   ;
}
```

would produce:

```
i is now 1
i is now 2
i is now 3
*
*
i is now 10
```

LabView for Loop


Untitled 1 Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
N
i

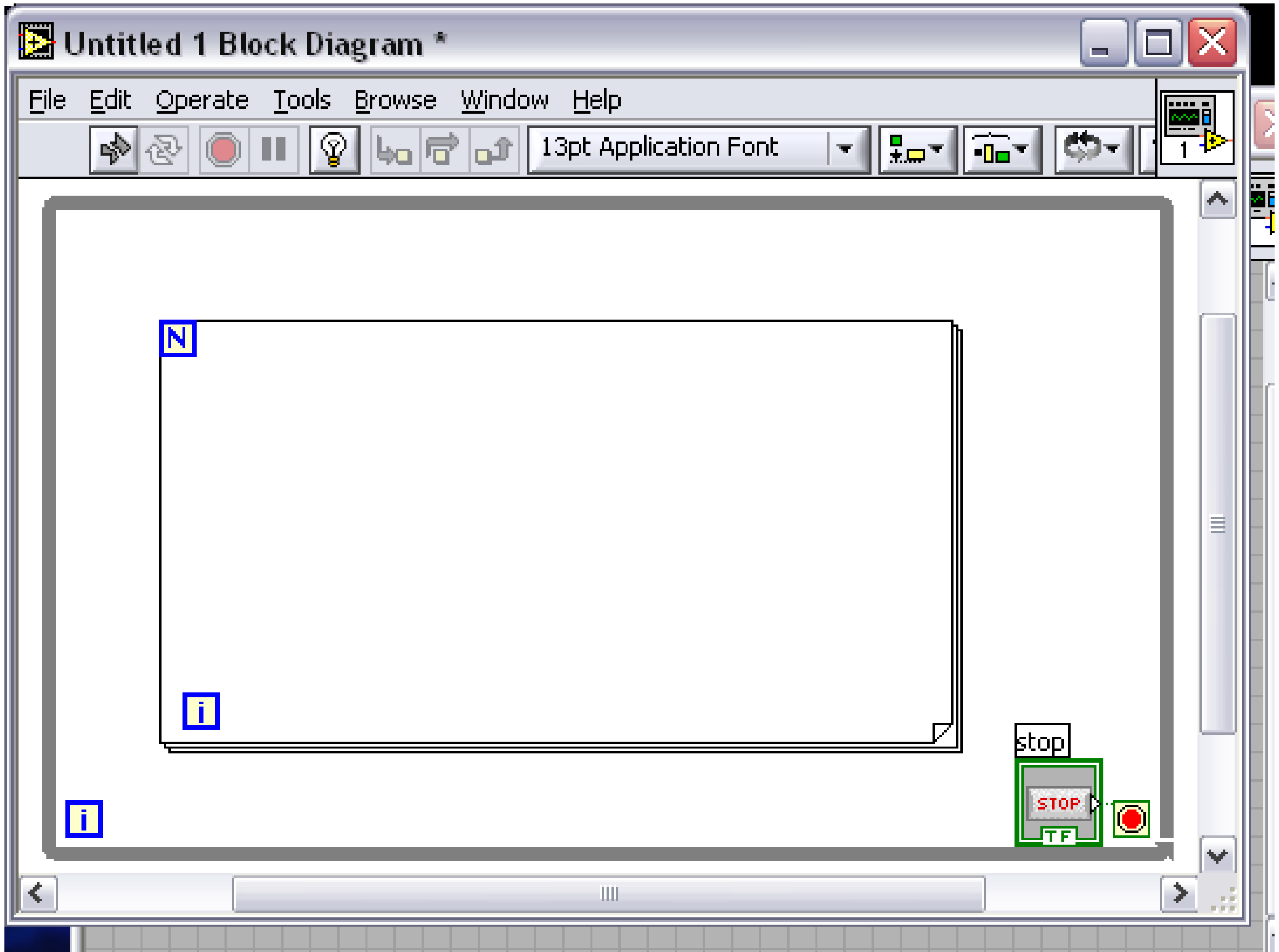
Repeat whatever VI's are in the box until the Boolean function (based on i and N) is true/false.

while/for Nested Loops

In either LabView or C programs, clarity is greatly enhanced by nesting a for loop inside of a while loop – for example to work on three items on many invoices you might find code like this:

```
while (1) {
        for ( i = 1; i < 4; ++i) {
                Do your business;
        }
}
```

LabView while/for nests


Untitled 1 Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
N
i
i
stop
STOP
TF
1

if (*Boolean condition*) statement;

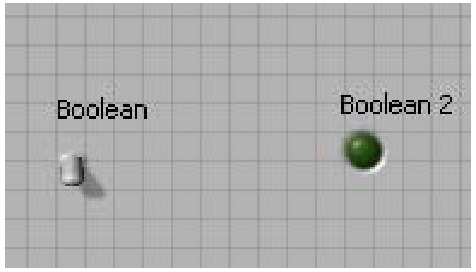
```
if (i  > 3) {
```

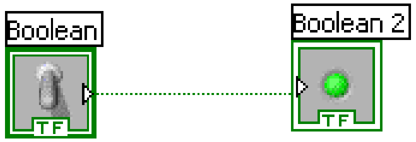
A compound statement is a bunch of statements enclosed by curly braces!

```
}
```

- A Boolean condition is either true or false.
- The statement (or sequence of statements) will be executed if the Boolean condition is true (1).
- Program execution will jump to the statement following the closing curly bracket if the Boolean condition is false (0).

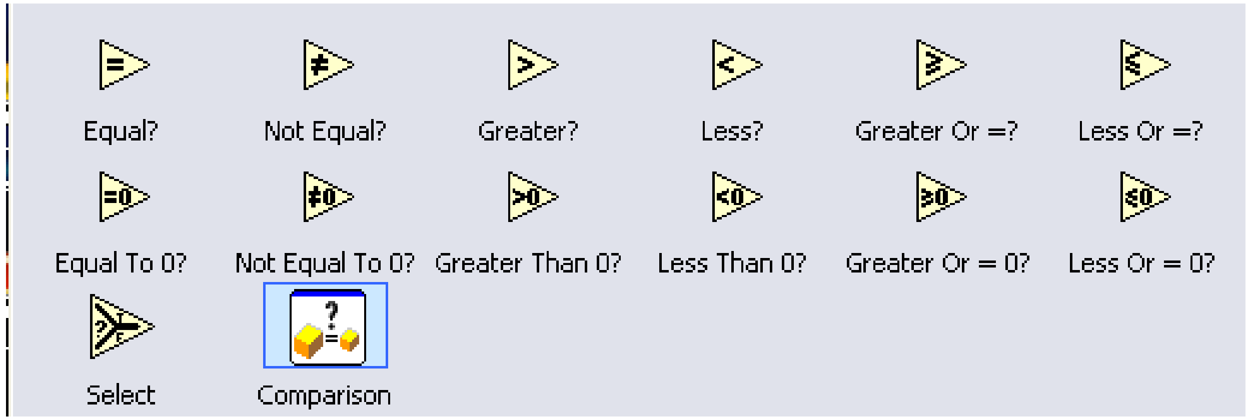
Boolean Conditions in LabView



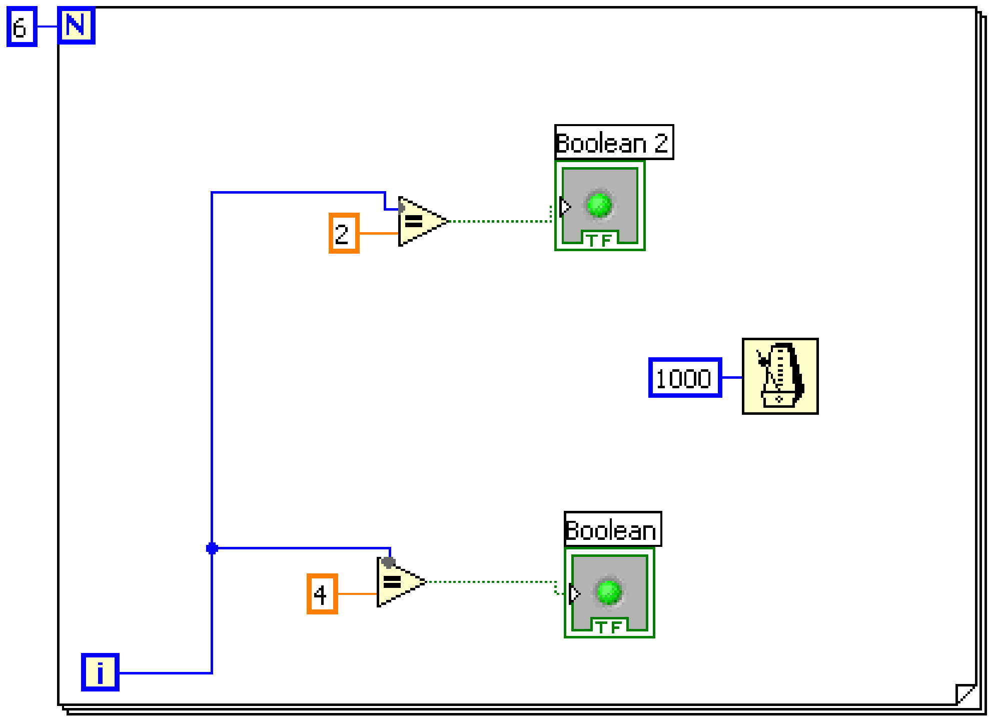


- If the switch is off, the LED remains dark
- If the switch is on, the LED lights up.
- Note that the ‘wire’ is faint green – This indicates transmission of a Boolean value.

Numerical Conditions in LabView



- A Boolean output variable shows whether the condition is true or false.



- When i is equal to 2, the upper LED lights up
- When i is equal to 4, the lower LED lights up.
- Otherwise, both LEDs are dark.
- The blue wires denote integers.

Arrays

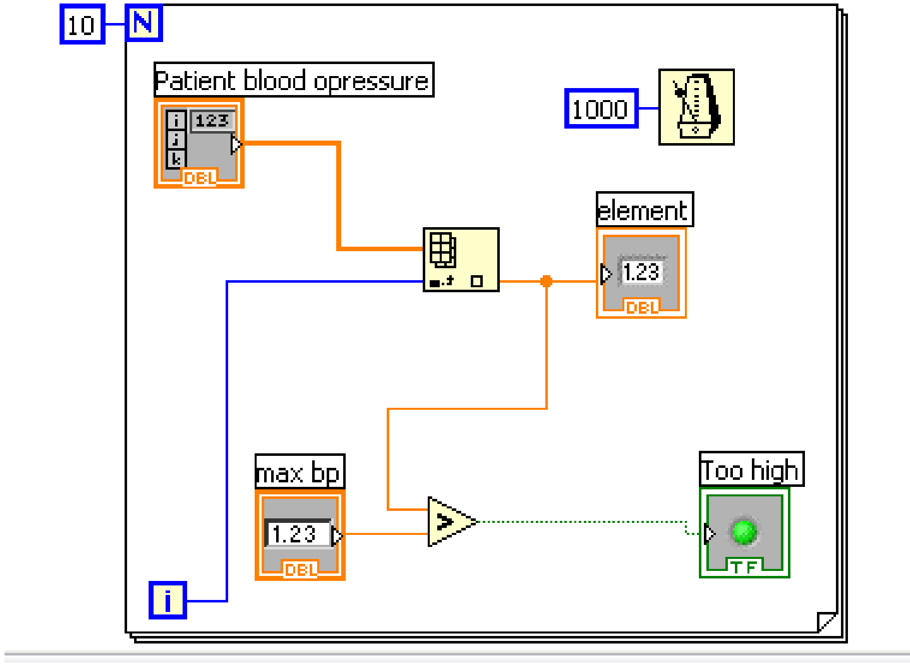
It is often useful to declare variables in a block – one entity is described by several numbers. For example, suppose the price of Dell stock, on the previous 100 business days, was stored in an array named dell[].Yesterday's price is in dell[99] –the day before is in dell[98], etc.A program to find the stock's peak might look like the following – *note the interplay between the index in the for loop And the index of the array dell.*

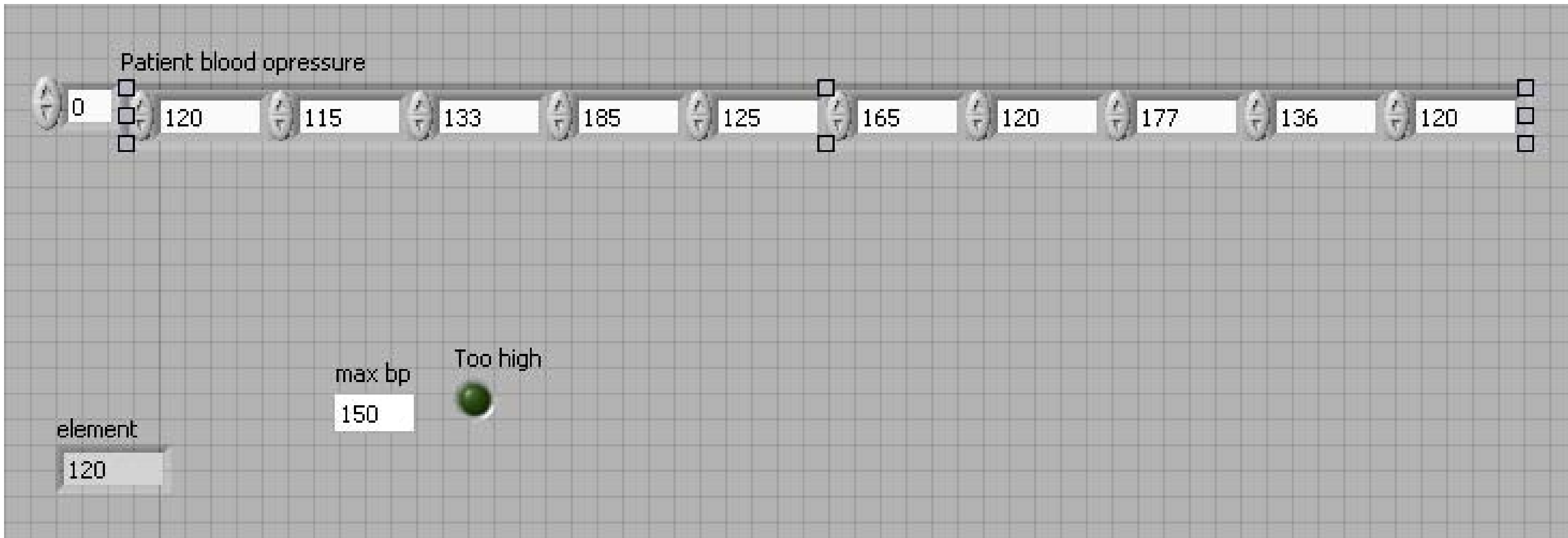
```
i = 0                            ;
peak = 0                          .;
for(i=0;i < 100;++i) {
    if (dell[i] > peak) {
            peak = dell[i]         ;
    }
}
```

LabView bp scan

LabView steps through the bp array. Readings higher than 150 cause the LED alarm to light up.

*Note the blue wire (integer),orange wire (floating point), green wire (Boolean)

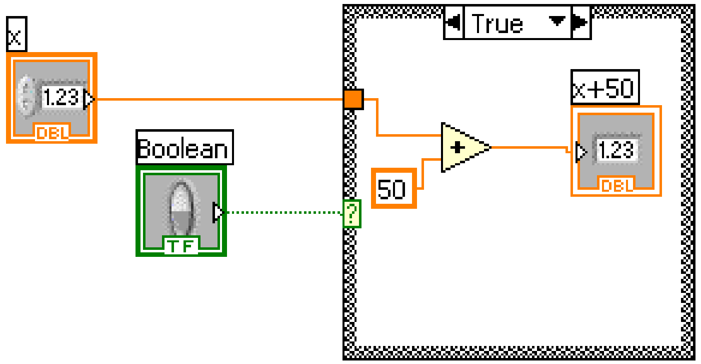


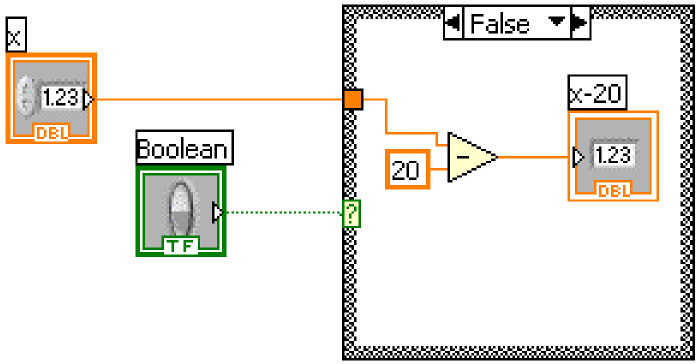


if (*Boolean condition*) statement;
else statement;

```
if (i  > 3) {
        If Boolean is true, do this;
}
else {
        If Boolean is false, do this;
}
```

LabView Case Structures





- If Boolean is true, perform operations in true structure.
- else, if Boolean is false, perform operations in false structure.