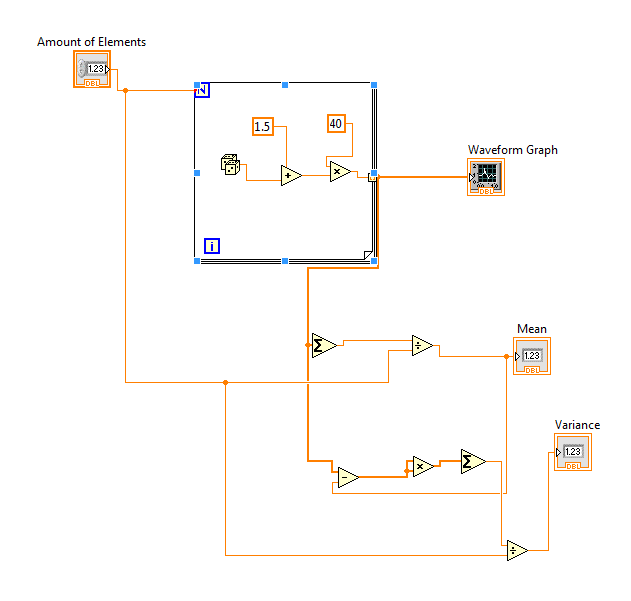
**Problem 1**

In the first problem, we were asked to create a user specified array of random numbers with values ranging from 60 – 100 and compute the mean and variance of this array. To do this I used a waveform graph and a numeric input for the user. For the output I used two numeric textboxes and made them display the mean and variance. When creating the random numbers we only wanted numbers 60 – 100. To do this we added 1.5 to the random number and then multiplied by 60.

A(X + B) | X = 0 🡪 = 60

A(X + B) | X = 1 🡪 = 100

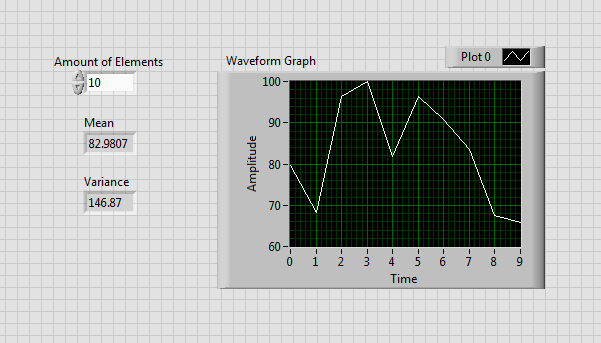
Solving for A and B, we get that A = 40 and B = 1.5

Adding 1.5 and multiplying by 60 gave us the desired range that we wanted for the random numbers.

To find the mean, we just used a “Sum Array Elements” component to add everything together and then divided by the number of elements. This was pretty easy, but the next part was a bit harder as we needed to compute the variance.

To compute the variance, we took the current random number and subtracted the mean from it and squared the result. This produced a positive number either way which allows us to easily add them up. To add all the numbers up, we again used a “Sum Array Elements” component and divided by the amount of random numbers generated.

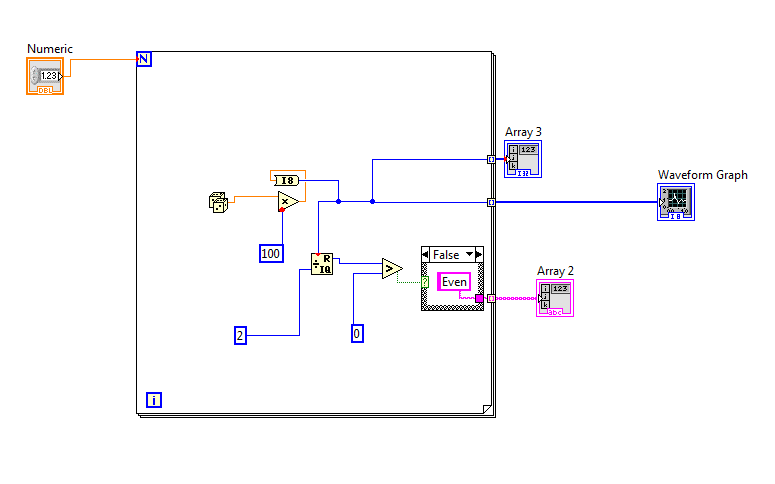
**Result:**

The following picture shows the result, a waveform graph that displays the random number generated at certain iteration. The picture also shows the amount of elements entered and the calculated Mean and Variance indicators.

**Problem 2**

In the second problem, we were asked to build a VI that uses a loop to generate an array of numbers with values ranging from 0 – 100 and indicate which numbers are even and which ones are odd. This array was also user sizeable. To do this, I did the same number generation as the first problem, but instead of adding 1.5 and multiplying by 60, I only multiplied by 100 and then converted the number to a discrete value. After that, I modded the result by 2 to find out if the number was even or odd. If the number is divisible by two (remainder is 0) then it is even. If it is not divisible by two (remainder is 1) then it is odd. I then took that result and put it into a greater than to test whether or not it was greater than 0, which then went into a True/False control block and printed out Odd/Even respectively. This printed result went into a string array and was displayed for the user to see. I also, for debugging purposes, included an array of all the numbers generated to cross check myself and mke sure the program was correctly computing the type. Also, just for fun, I included the waveform graph like we did last time. This was mainly for me as I wanted to see the distribution of numbers to determine how random the set of numbers really were.

**Program code:**



**Result:**

The picture below shows the result of running the program for a user specified 10 integer array. The waveform displays an interesting up-down distribution while Array 3 is displaying the numbers generated and Array 2 is displaying the type of number (Even/Odd) that the program detected it as.

