Homework:

We will continue developing our 1D array class. **However, the elements in this array class will now be 2x2 arrays.** This means that when you allocate memory for your vector elements, you must also allocate memory for the 2x2 arrays. The 2x2 arrays contain integers only.

1. Constructors. Provide several constructors which provide different ways of creating objects. Be creative.
2. The '=' operator (copy assignment operator), which copies one array to another array.
3. The '*' operator, which multiplies two arrays, element by element (elements of the same index.) Do not forget to check the two arrays have the same size. If they are of different sizes, print out an error message and return the left array.
   For example, in the last homework, Array (1, 2, 3) * Array (2, 3, 4) = Array(1*2, 2*3, 3*4) = Array (2, 6, 12) . In this homework, 1*2 gets replaced by the matrix-matrix multiplication of (2x2) arrays. So: Array(a, b, c) * Array(d, e, f) = Array(a*d, b*e, c*f), where a*d is the matrix-matrix multiplication of 2x2 by 2x2 matrices.
4. The '*' operator (using the friend concept) to multiply an array by a float (or double). Thus, if "a" and "b" are arrays, and "f" is a float, the following should work: "b = f * a" and "b = a * f". Hint: you will need two functions to implement this. You will need to multiply each element of the 1D array by the float. Therefore, you must multiply 2x2 matrices by a float: each matrix element is multiplied by this float.
5. The '[]' operator, retrieve the specified element of the array, which now will return a 2x2 matrix.
   For example: if "Array a(…);" has been defined, "a[3]" retrieves the 3rd element of the array, which is a 2x2 matrix.
6. The '%' operator to encode the dot product (recall, the dot operator cannot be overloaded). Thus, if "a" and "b" are arrays, " a % b" should return the sum of a[i]*b[i] over its elements, and return a float or double (up to you.) Again, in this problem, a[i]*b[i] are matrix-matrix multiplication since a[i] and b[i] are 2x2 matrices.
7. The '<<' output the array, which allows "cout" to be used on the array.

Create a main function to test your overloaded operators. Each function should have at least two tests.

Zip your code, with a readme file and a Makefile and submit to Canvas.