Honors Computer Science C++
Mr. Clausen
PROGRAM 10A
apmatrix

PROGRAM 10A Matrices not Maître d's (25 points)

This program is designed to give you practice with the apmatrix class. It is a "safe" two dimensional array based upon the apvector class.
Before the menu begins, in the int main function, ask the user how many rows and columns they would like in their matrix. Error trap these numbers to make sure they are not negative numbers. Also ask the user what fillValue they would like to use to initialize the matrix. Create this matrix named matrix, to store integers, and create another matrix named transposed, to contain the transposed values of this matrix. Don't actually transpose the matrix in the int main just create the matrix. You will transpose the matrix in another function.
The transpose of a matrix is a new matrix with the row and column positions reversed. The transpose of a matrix called matrix which has M rows and N columns (an M x N matrix: that's M by N) is an N x M matrix called transposed, with each element matrix[M] [N] being stored in transposed[N] [M].
This program will also test to see if the matrix is symmetric. A matrix is symmetric if it has the same number of rows and columns (called a "square" matrix) AND if each element matrix[M] [N] is equal to matrix[N] [M]. By the way, a symmetric matrix will be the same as it's transposed matrix.
Create functions for: a Menu, a Control of the Menu execution, a function to fill the matrix with random numbers, a function to test for symmetry, a function to display the original matrix, a function to display the transposed matrix, and a function to create another matrix (actually you cannot recreate the matrix, just resize matrix) and allow the user to enter data values for each element in the matrix.
Below is a sample of what the program run and Menu might look like:

```
Enter the number of rows you wish to use in your matrix: 3


Enter the number of columns you wish to use in your matrix: 5


What number would you like to use to initialize the matrix? -9


           Main Menu


1.  Fill the matrix with random integers.
2.  Create another Matrix and Enter the elements.
3.  Test the matrix for symmetry.
4.  Transpose the matrix.
5.  View the elements in the original matrix
6.  View the elements in the transposed matrix
Q.  Quit the program


Enter your Choice:
```

Let's look at examples of transposed and symmetric matrices…

Let's start with the following 2 x 3 matrix as our original matrix, named matrix.

matrix: a 2x3 matrix.

1	2	3
4	5	6

The transposed matrix named transposed, will be a 3 x 2 matrix with the row and column elements reversed. The element at index number [0] [0] will stay there, since reversing two zeros does not change the position of the element. The element at index number [0] [1] will be relocated to index position [1] [0]. The element at index number [0] [2] will be relocated to index position [2] [0], etc.

transposed: a 3 x 2 matrix:

1	4
2	5
3	6

A symmetric matrix must have the same number of rows and columns, and transposed must be equivalent to itself. Here is an example of a symmetric matrix, use this as test data for your program.

matrix: a 3 x 3 symmetric matrix.

4	1	2
1	5	7
2	7	6

You can see that index numbers [0] [0], [1] [1], and [2] [2] will not move when we switch the row and column numbers (the diagonal going down from left to right).
The element "1" at index [0] [1] must be the same as the element at index [1] [0], the element "2" at index [0] [2] must be the same as the element at index [2] [0], etc. In other words, if you were to "fold" the matrix across the diagonal line, the elements must match each other. You are already familiar with graphs symmetric to a line from previous math classes, and you should be familiar with matrices from your Algebra II class.

Enjoy!