HW 11

Write a matrix multiply A = B x C that first uses 1, then 2 and then 4 threads. Let N be the number of rows and columns in A, and let N be divisible by 2. Your program should:

1. Read the size of A (i.e. N where A, B and C are all NxN matrices)
2. Call a function *fill(int N)* that fills in B and C with whatever numbers you want to use. *Do not* read them in, however.
3. The *fill* function should throw an exception if N % 2 != 0, i.e. if N is not divisible by 2
4. If thrown, the exception should be caught in the method that calls *fill*. The catch clause should print out the erroneous value of N that has been stored in the exception and prompt for a new value. It should the re-call *fill*. This should be done until a valid value of N is received.
5. After filling the array, perform the multiply with 1, 2 and 4 threads. With one thread the entire matrix A will be computed by the single thread. With two threads the top 1/2 of the rows will computed by thread 0 and the bottom half by thread 1. With four threads the upper left quadrant of A will be computed by thread 0, the upper right by thread 1, the lower left by thread 2 and the lower right by thread 3.
6. The time taken to perform each matrix multiply should be found and printed. You will need to use fairly large matrices (10000 x 10000?). HW10 shows how to access the time in Java.