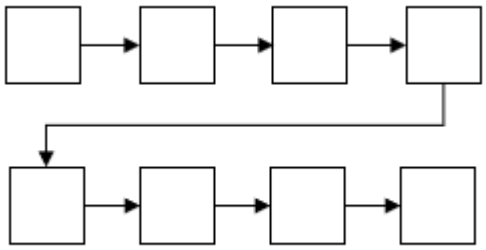
Homework 3 – Due Mar. 27th 23:59, KST

Instructions: Complete the implementation and turn it in before the due date. Any deviations from the instructed deliverable format will result in a deduction of grade. DO NOT COPY OTHER'S WORKS!

In this assignment, you will be implementing an integer matrix using an array list. The matrix is represented as a chain (i.e., a 1D list) that is "bent" several times to the shape of a rectangle. For example, a 2x4 matrix, having two rows and four columns, is represented as the following list (arrows are shown to indicate the next item):



Here, the first element of the list corresponds to the element in the first row and first column of the matrix. The second element corresponds to the first row and second column, etc. and so on. This kind of format is known as a 'row-major order' and is a popular choice among many programming languages to store 2D arrays.

Your task is to fill in the methods provided in ALMatrix.java so that the class can perform trivial data management operations in row-major order. Carefully read all comments given in ALMatrix.java before proceeding with the homework. The instructions in the comments are also part of the official requirements.

- Documentation (40 points): You should provide a header comment that provides a big-O time complexity analysis for each of the required methods. Notice that I'm not providing the variable of complexity: You must clearly identify with respect to what variable the time complexity will be. In addition to the big-O's, provide a brief explanation of how you arrived at that conclusion.
- Correctness (60 points): Your code should behave as required. We will use our own grading script to assess the correctness.
- Miscellaneous: Do not change the method and class names. Do not use a package structure. Two or more unhandled exceptions will result in a score of 0 for correctness.

Deliverable: A single ALMatrix.java source file. DO NOT provide a zip file. Just submit a single Java file with no package structure.