Recursion – Laplace Expansion

**Purpose**

This lab was designed to teach you how to think recursively.

**Description**

Write a function that accepts a square matrix (n x n 2D array) and returns the determinant of the matrix. The determinant of a matrix is a special number that is useful for solving systems of equations.

Case 1x1 matrix:

|a| has determinant a.

Case 2x2 matrix:

[ [a, b], [c, d] ] or

|a b|

|c d|

has determinant: a\*d - b\*c.

The determinant of an n x n sized matrix is calculated by reducing the problem to the calculation of the determinants of n matrices of n-1 x n-1 size.

Case 3x3 matrix:

[ [a, b, c], [d, e, f], [g, h, i] ] or

|a b c|

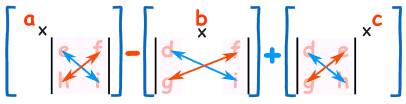
|d e f|

|g h i|

the determinant is:

a \* det(a\_minor) - b \* det(b\_minor) + c \* det(c\_minor)

where det(a\_minor) refers to taking the determinant of the 2x2 matrix created by crossing out the row and column in which the element a occurs:

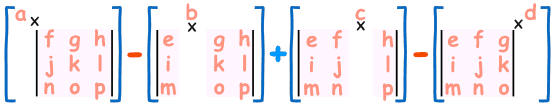


Note the alternation of signs.

The determinant of larger matrices is calculated analogously, e.g. if M is a 4x4 matrix with first row [a, b, c, d], then:

det(M) =

a \* det(a\_minor) - b \* det(b\_minor) + c \* det(c\_minor) - d \* det(d\_minor)



And the pattern continues for 5x5 matrices and so on. The Laplace expansion is an expression for the determinant of a n x n matrix that is a weighted sum of the determinants of n sub-matrices or minors each of size n – 1 x n – 1. For large matrices, this expansion quickly becomes inefficient and other methods should be used.

**Program Shell**

Create your own.

**Sample Execution**

determinant(new int[][]{{2, 5, 3}, {1, -2, -1}, {1, 3, 4}})

// should return -20

determinant(new int[][]{{1, 3}, {2, 5}})

// should return -1

determinant(new int[][]{{3}}

// should return 3

determinant(new int[][]{{5, 3, 2, 5, 3}, {31, 5, 81, -2, -1},{0, 3, 1, 3, 4}, {0, 8, 5, 3, 3}, {-1, 3, -8, 3, 3}})

// should return 9554

determinant(new int[][]{{1,2,3}, {4,5,6}, {7,8,9}})

// should return 0(the matrix isn’t invertible and either has no solutions or infinitely many)

determinant(new int[][]{{5, 3, 2, 5, 3, 1}, {1, 5, 1, -2, -1, 2},{1, 3, 1, 3, 4, 1}, {0, 8, 5, 3, 3, 0}, {-1, 3, 4, 3, 3, 2}, {4, 3, 1, 3, 3, 2}})

// should return 621