**Two Dimensional Array Problems**

1. Given two matrices, A and B, produce a matrix C, which will hold the summed values of A and B.

Assume that both A and B are m X n, where m is the row size and n is the column size.

Write an algorithm in ADL as a black box to calculate C, i.e.

procedure MSum(IN A[][], In rowSize, IN columnSize, B[][], OUT C[][])

You can get more information from <https://www.mathsisfun.com/algebra/matrix-introduction.html>

1. Given two matrices, A and B, produce a matrix, C, which will hold the multiplied values of A and B.

Assume that A is m X n, B is n X p. Hence, C = A \* B, an m X p matrix. Write an algorithm in ADL as a black box to calculate C, i.e.

procedure MMult(IN A[][], In m, IN n, B[][], IN p,

OUT C[][])

You can get more information from <http://www.mathsisfun.com/algebra/matrix-multiplying.html>

1. An m X n matrix is said to have a *saddle* point if some entry A(i, j) is the smallest value in row I and the largest value in column j.

Write an algorithm in ADL which determines the location of a *saddle* point if one exists. (Hint: you can, if you wish, create two abstractions called MinInThisRow, which will return the minimum value in a given row of a two dimensional array, and MaxInColumn, which will return the maximum value of a given column of a two dimensional array.