Our Flounder

Computer Science Correspondence School

discere domi

Lecture 25

CISC 131
Introduction to Programming and Problem Solving
Spring 2020
Array Lab

Due: Friday, May 15, 2020, at start of class.
Points: None

In this lab, you will write some functions that do matrix arithmetic. A matrix is represented as a two dimension array. You might want to do a little research to help refresh your memory about how matrix arithmetic works. A few of the functions that you will need for this lab have already been done in previous labs (i.e., *isRectangular*, *sameShape*) so just copy those into this program as you need them but make sure they work correctly before you do that. Test thoroughly with small matrices so you can check to ensure your functions are working..

Problem 1
Write a function with this function header:

```
function canBeAdded ( array2D1, array2D2 )
```

The function is passed two two-dimension arrays and returns *Boolean true* when the actual parameters are two dimension arrays for which matrix addition is defined and *Boolean false* if they cannot be added. You may want to refresh your memory on the requirements for doing matrix addition.

Problem 2
Write a function with this function header:

```
function canBeMultiplied ( multiplicand, multiplier )
```

The function is passed two two-dimension arrays and returns *Boolean true* when the actual parameters are two dimension arrays for which matrix multiplication is defined and *Boolean false* if they cannot be multiplied. You may want to refresh your memory on the requirements for doing matrix multiplication.

Problem 3
Write a function with this function header:

```
function add ( array2D1, array2D2 )
```

The function is passed two, two dimension arrays. If the arrays represent matrices that can be added, the function returns a two dimension array that represents the result of doing matrix addition on the formal parameters. If the arrays do not represent matrices that can be added, the function returns a zero length array.

Problem 3
Write a function with this function header:

```
function subtract ( minuend, subtrahend )
```

The function is passed two, two dimension arrays. If the arrays represent matrices that can be subtracted, the function returns a two dimension array that represents the result of doing matrix subraction on the formal parameters. If the arrays do not represent matrices that can be added, the function returns a zero length array.

Problem 4

Write a function with this function header:

```
function transpose ( array2D )
```

Write a function is passed a two dimension array that represents a matrix. The function returns a two dimension array that represents the transpose of the formal parameter matrix.