Inlab 5: MultiDimensional Arrays

You are expected to turn this in to your lab instructor at the start of your next lab period. It is worth approximately 0.5% of your course grade. Recall that you will be allowed to drop two in-lab assignments.

You are to turn in a printout of your code as well as a printout of the result of running the code (cut-and-paste from the screen into the editor). You are to staple these two printouts together and insure that your name, section number and unique name are visible at the top of the front sheet. ***Be sure your name, section number, and uname are at the top of the code (as a comment!)***

Problem 1: Printing the matrix (28 points)

The program md_arrays.cc makes an identity matrix of size ROW x COL, and prints it out. The main function fills the matrix with the appropriate numbers, and calls the function print_2Darray. The function print_2Darray will print out that matrix row-by-row with spaces between numbers and a newline at the end of every row. For example:

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

Your job is to write the function print_2Darray and make sure it works.

Problem 2: An identity matrix (2 points)

Modify the loop that populates the matrix within the main function. Instead of creating a diagonal of ones that begins in the upper-left you will now produce a diagonal that starts in the upper right. Once completed, your program will use the print function from Problem 1 to print this new matrix and only this new matrix. (We should no longer see the identity matrix from Problem 1).

```
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0
```

Things to think about:

1. The main function calls the print_2Darray function at the bottom; use that to help you define the print_2Darray function.
2. Remember, there's something tricky about declaring a 2-D array as a function parameter.
3. Think about how you iterate through a regular array. How do you extend that to two dimensions?
4. Think about printing to screen. When things are printed out, the first row is printed from left to right, and then the next row is printed from left to right, and then the next...etc.
5. Putting cout's in your code is a good way to see what it's doing.
6. The other code for *filling* the 2-D array (in main) should be very similar to code for *printing out* a 2-D array (except printing out is a bit simpler).
7. In Prob. 2 it will be helpful to write out the <x,y> coordinates you will need to fill with ones and look for a pattern. The code within the while loop can be adjusted to exploit this pattern.