Assignment on Pointers in Functions:

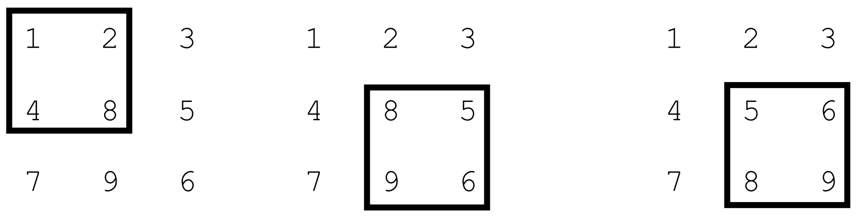
You are to submit the source code (printout and softcopy). For the softcopy, bring a copy of the source code in USB. Ideally, you also have another back-up copy (maybe sent to YOUR OWN email account).

Rotation

Kassandra used to own an old Nokia phone. In that phone, she remembered playing a game called Rotation, which involved rotating a grid of numbers to return them back to their proper orientation. Games start with the board in disarray, and the game would end only when all the numbers on screen are in their proper place.

A player chooses to move a box encompassing four numbers, and then chooses to either rotate the numbers clockwise or counterclockwise. A sample game grid and a valid move is show below:

Start position **Move Box** **Rotate counterclockwise**



Your ONLY task is to create a simple program simulation of the clockwise and counterclockwise rotation in a **2x2** (shown above is a 3x3 board) board. **Create one function that could simulate the clockwise and another function to simulate the counterclockwise rotation.** For the interest of this problem, you are to begin with a board in the proper orientation (i.e. 1 2 for the first row and 3 4 for the second row.)

Take note that we are interested only in the rotations, and not winning or losing (i.e. you do not have to check if the grid is in the proper position).

Your program should ask if a clockwise or a counterclockwise rotation is wanted. You are then to display the board orientation. Ask 1 more time if a clockwise or a counterclockwise rotation is wanted, then display the result. Follow the sample screen outputs show below.

```
Sample 1:
*********************************************************
Press 1 for clockwise and 2 for counterclockwise: 1
   3   1
   4   2
Once more: Press 1 for clockwise and 2 for counterclockwise: 1
   4   3
   2   1
*********************************************************

Sample 2:
*********************************************************
Press 1 for clockwise and 2 for counterclockwise: 2
   2   4
   1   3
Once more: Press 1 for clockwise and 2 for counterclockwise: 1
   1   2
   3   4
*********************************************************
```

Take note that your instructor should be able to modify your code to either add or remove rotation moves. This means that the use of comments and practices for code readability is imperative. Your program should still be able to return the proper final board orientation, amidst the modifications.

Take note also that you are **not** allowed to put any `scanf()` or `printf()` in the clockwise and counterclockwise functions.

[Print this page. Your answer may be handwritten, but it has to be on the printout of this sheet.]

Name ____ Section ________________

Pointers
For this item, you will not be programming. Instead, you will answer the questions 1 – 5 in the next page. This is a paper and pen activity (i.e., you are to understand the effect of the code and not run it in the machine.

- **Research:** Operator precedence in C

The *binary operator* * is the multiplication operator. The *unary operator* * is the pointer/address. At what position in the operator precedence does the unary operator belong to? What is the direction of evaluation (associativity) for the unary operator *?

```
void fxnOne (int *pOne)
{
   (*pOne)++;
}

void fxnTwo (int *pOne)
{
   *pOne++;
}

int main()
{
   int nVal = 10;
   /* assume nVal is assigned to memory address  3000 */
   fxnOne (&nVal);

   fxnTwo (&nVal);
   return 0;
}
```

Answer the following questions:

1. What is the value of `nVal` in `main()` after execution of each function call?
2. What is the value of `pOne` before and after execution of the statement in `fxnOne()`?
3. What is the value of `*pOne` before and after execution of the statement in `fxnOne()`?
4. What is the value of `pOne` before and after execution of the statement in `fxnTwo()`?
5. What is the value of `*pOne` before and after execution of the statement in `fxnTwo()`?