Lab 7: Pointers in C

Reading: Chapter 7 - Pointers
Also, chapter 8 – C Characters and Strings is very helpful

All of your code will be in one file, lab7.c.

1. Create a C program named lab7.c. Include the header files stdio.h, stdlib.h, and time.h then begin the main function.

2. In the main function create a one-dimensional integer array with 80 elements. (Note: This lab intentionally uses a one-dimensional array.) Fill the array with random integers between 1 and 1000.

3. Print an informative message, then call the printTable function (see below) with this call:

```
printTable( array, 8, 10 );   // 8 rows, 10 columns
```

4. Call the columnSort function (see below) with this call:

```
columnSort( array, 8, 10 );
```

5. Print an informative message, then call the printTable function again.

```
printTable( array, 8, 10 );   // 8 rows, 10 columns
```

Let's look at an example with a smaller array. Suppose that I create an array with 12 elements and fill it with random numbers. After this filling process, I have the array:

Index	0	1	2	3	4	5	6	7	8	9	10	11
Value	78	14	283	419	6	794	91	52	89	511	386	29

Now, I call the printTable function:

```
printTable( array, 3, 4 );
```

I expect to see the following output:

```
 78   14  283  419
  6  794   91   52
 89  511  386   29
```

Now, I call the columnSort function and the printTable function again:

```
columnSort( array, 3, 4 );
printTable( array, 3, 4 );
```

I expect to see the following output:

```
  6   14   91   29
 78  511  283   52
 89  794  386  419
```

The numbers within each column are sorted. The smallest value in the column is at the top.

Use Incremental Development. Comment the line that calls columnSort while you work on printTable.

6. Before the main function, place the prototype for printTable.

```
void printTable( const int * const array, const int rows, const int columns );
```

7. After the main function, place the function definition for printTable.

   - You may use either pointer/offset notation or pointer/index notation but you must be consistent.
   - Use a field width of 5 for printing each integer.
   - Do not use the tab character '\t'.

8. Compile, run, debug if needed, repeat until printTable works correctly.

Use Incremental Development. Uncomment the line that calls columnSort.

9. Before the main function, place the prototype for columnSort.

```
void columnSort( int * array, const int rows, const int columns );
```

10. After the main function, place the function definition for columnSort.

    - You may use any sorting algorithm you like.
    - You may not create any new arrays.
    - You may use either pointer/offset notation or pointer/index notation but you must be consistent.