60-141
Intro to Algorithms & Programming II Winter 2014

Lab #5: More on Pointers
(Due at the end of the lab period or beginning of the next)

Objective: Learn to use pointers and operators at a more advanced level.

A. Pointer Operators

Type the following program and document every line by explaining its function and output. If any line produces an error, comment that specific line and explain the nature of the error. Write your code for this part in a file called: **Lab5a.c**

```
#include <stdio.h>

int main()

{

int a = 7 ;

int *aPtr ;

aPtr = &a ;

printf( "%p", &a );

printf( "%p", aPtr );

printf( "%p", &aPtr );

printf( "%d", a );

printf( "%d", *aPtr );

printf( "%p", *&aPtr );

printf( "%p", &*aPtr );

printf( "%d", *&a );

printf( "%d", &*a );

return 0;

   }
```

B. Array Manipulation with Pointers

Write, document and test each of the following function specifications **(Lab5b.c):**
[You should use only pointer arithmetic and no array index].

1. A function called **FillArray()** that accepts a pointer to a integer as the array name, an integer for its size. This function randomly fills the array with integers ranging from 0 to 100.
2. A function called **PrintArray()** that accepts a pointer to a integer as the array name, an integer for its size. This function only prints the array elements.
3. A function called **BubbleSort()** that accepts a pointer to a integer as the array name, an integer for its size. The function should sort the array passed, in descending order. Use the function **Swap()** that you designed in the last Lab 4, to swap any values in the function **BubbleSort()**.
4. In the main() function:

    a. Declare an integer array called **NumList** of size SIZE (define SIZE as 20).
    b. Populate the array with random numbers [0 -100] by calling the function **FillArray()**.
    c. Display the Array contents by calling the function **PrintArray()**.
    d. Sort the array **NumList** in descending order by calling the function **BubbleSort()**.
    e. Display the SORTED array by calling the function **PrintArray()**.

For your convenience, the function prototypes are given bellow:

```
void FillArray ( int *array, int size );
void PrintArray ( int *array, int size );
void BubbleSort ( int *array, int size );
void Swap ( int *x, int *y );
```

EVALUATION:
You need to show your instructor the complete programs at the end of this lab, or at the beginning of your next lab. The marks you will receive for this lab are made of two parts: Lab work marks 8 and **attendance marks 2**. **Total 10 marks**.

Lab Work Mark: You will be evaluated based on your solutions for the problems based on the following scheme:

0 mark = No work done.
2 mark = Incomplete code / does not compile, with no/invalid documentation
4 marks = Complete running program with no/invalid documentation
6 marks = Incomplete code / does not compile, with proper documentation
8 marks = Complete running program with proper documentation

IMPORTANT:
ASK QUESTIONS IF YOU GET STUCK, BUT DO YOUR OWN CODE. ANY CODE SUSPECTED TO BE SIMILAR TO ANOTHER SUBMISSION WILL CAUSE BOTH SUBMISSIONS TO RECEIVE A ZERO MARK ON ALL LABS AND BE REPORTED FOR PLAGIARISM