Due: April 26, 11:59pm

Objective: Pointers and Arrays.

Submit the source code through Canvas. *yourname_Lab8.c* is the file name. Use proper commenting at the beginning of your code with your name, lab number and date.

Use void functions for each problem unless stated otherwise.

(Ten points deduction for not following any of the given formats)

1- (8 point) Answer each of the following. Assume that integers are stored in 4 bytes.

a) Define an array of type int called `apples` with five elements, and initialize the elements to the even integers from 2 to 10. Assume the symbolic constant `SIZE` has been defined as `5`.
b) Define a pointer `aPtr` that points to a variable of type int.
c) Print the elements of array values using array subscript notation. Use a `for` statement.
d) Give two separate statements that assign the starting address of array values to pointer variable `aPtr`.
e) What address is `aPtr` pointing to?
f) Print the elements of array values using pointer/offset notation.
g) What address is referenced by `aPtr + 3`? What value is stored at that location?
h) Assuming `aPtr` points to `apples[4]`, what address is referenced by `aPtr -= 4`? What value is stored at that location?

2- (4 points) Do each of the following in the `main` function:

a) Write the function prototype for function `zero`, which takes a double array parameter `bigNums` and does not return a value.
b) Write the function call for the function in part a.
c) Write the function prototype for function `add1AndSum`, which takes an integer array parameter `oneTooSmall` and returns an integer.
d) Write the function prototype for the function described in part c.

3- (8 points) For each of the following, write a statement that performs the indicated task. Assume that floating-point variables `number1` and `number2` are defined and that `number1` is initialized to `7.3`.

a) Define the variable `fPtr` to be a pointer to an object of type double.
b) Assign the address of variable `number1` to pointer variable `fPtr`.
c) Print the value of the object pointed to by `fPtr`.
d) Assign the value of the object pointed to by `fPtr` to variable `number2`.
e) Print the value of `number2`.
f) Print the address of `number1`. Use the `%p` conversion specifier.
g) Print the address stored in `fPtr`. Use the %p conversion specifier. Is the value printed the same as the address of `number1`?