CS1050 – Lab 7
Fall 2021

Concepts to Practice
- Simulated "pass by reference" via pointers
- Relationship between arrays and pointers
- Expand the prelab

Submission Information
Submit this assignment by following the instructions below. SUBMIT ONLY the .c file (no a.out or executable file is required).

Use the following submit command:

mucs submit <class> <assignment_type> <filename>

For example:

mucs submit 1050 lab7 lab7.c

Description
For the lab assignment, you need to write four functions in addition to main() based on the prototypes shown below. The goal is to have the user enter values into an array of integers, print that array out, print out the total of all elements in the array, and print out the average of all elements in the array.

Your main() function will only:

- Declare any variables whose values will be entered by the user (you may declare other variables in main() as well if you wish).
- Print out a welcome message.
- Call functions that you have written.
- Print out the total.
- Print out the average.

Your main() function may not:

- Call scanf()

Functions
You may not use global variables (using global variables will result in a zero). To get full credit, you should write functions that correspond to the following prototypes:

- **void GetIntArray(int *outputArray, int * count)** – This functions prompts the user for how many integers they wish to enter, and then prompts the user for each integer to be put in the array.
- **void PrintArray(int * array, int count)** – This functions prints out count elements of the array.
- **int TotalArray(int * array, int count)** – This function sums the first count elements of the array and returns this total.
- **float AvgArray(int * array, int count)** – This function averages the first count elements of the array and returns this average.

Honors Extension

For the honors extension, you are to use your GetIntArray() function to input a 2nd array in addition to the array mentioned above. After the user enters the values in this array, print the total of both arrays (the sum of the elements of the 2 arrays) and the average of both arrays (the average of all of the elements of the 2 arrays together).

Non-honors Sample Output1

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab7-8$ ./a.out

************************
* Welcome to Lab 7-8    *
************************

Enter the number of elements in the array:
5
Enter element #0:1
Enter element #1:2
Enter element #2:3
Enter element #3:4
Enter element #4:5

Here are the 5 elements of your array:
        Element 0 = 1
        Element 1 = 2
        Element 2 = 3
        Element 3 = 4
        Element 4 = 5
Total of array = 15
Average of array = 3.00
```

Non-honors Sample Output2

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab7-8$ ./a.out

************************
* Welcome to Lab 7-8    *
************************

Enter the number of elements in the array:
3
Enter element #0:15
Enter element #1:25
Enter element #2:33

Here are the 3 elements of your array:
        Element 0 = 15
        Element 1 = 25
        Element 2 = 33
Total of array = 73
Average of array = 24.33
```

Honors Sample Output1

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab7-8$ ./a.out

************************
* Welcome to Lab 7-8    *
************************

Enter the number of elements in the array:
5
Enter element #0:1
Enter element #1:2
Enter element #2:3
Enter element #3:4
Enter element #4:5

Here are the 5 elements of your array:
        Element 0 = 1
        Element 1 = 2
        Element 2 = 3
        Element 3 = 4
        Element 4 = 5
Total of array = 15
Average of array = 3.00
Enter the number of elements in the array:
4
Enter element #0:1
Enter element #1:2
Enter element #2:3
Enter element #3:4
Total of array and array2 = 25
Average of array and array2 = 2.78
```

Honors Sample Output2

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab7-8$ ./a.out

************************
* Welcome to Lab 7-8    *
************************

Enter the number of elements in the array:
3
Enter element #0:15
Enter element #1:25
Enter element #2:33

Here are the 3 elements of your array:
        Element 0 = 15
        Element 1 = 25
        Element 2 = 33
Total of array = 73
Average of array = 24.33
Enter the number of elements in the array:
2
Enter element #0:100
Enter element #1:200
Total of array and array2 = 373
Average of array and array2 = 74.60
```

Guidelines for Grading Lab 7
40 Points Possible

General

If your program does not compile or produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of ZERO POINTS. Further, if your program does not actually follow the specifications, but merely prints out lines that make it appear to follow the specifications, you will receive a grade of ZERO POINTS. For partial credit your C program must not only compile but also produce some valid I/O that meets the lab specifications.

You program is expected to have a comment header at the top that includes your name, pawprint, the course you are taking, and the lab that you are solved (e.g., "Lab 7"). Your program should also include appropriate comments where needed. Your code should be nicely indented. **You will lose up to 10 points if you do not meet these basic requirements**.

If you do **ANYTHING** in your main() other than declaring variables, calling printf() to output the welcome message, or calling one of the specified functions that you will write, you will lose 50% of the points you would have otherwise received. If you have any global variables, you will get zero points.

Non-honors Rubric

2 points: GetIntArray() correctly prompts for and receives a count.
3 points: GetIntArray() correctly prompts for count elements in an array.
5 points: TotalArray() correctly sums the relevant elements in the array and returns this sum.
5 points: AvgArray() correctly averages the relevant elements in the array and returns this average.
5 points: PrintArray() correctly prints the relevant elements in the array.
5 points: GetIntArray() works correctly and does not use the index [] operator.
5 points: TotalArray() works correctly and does not use the index [] operator.
5 points: AvgArray() works correctly and does not use the index [] operator.
5 points: PrintArray() works correctly and does not use the index [] operator.

Honors Rubric

2 points: GetIntArray() correctly prompts for and receives a count.
3 points: GetIntArray() correctly prompts for count elements in an array.
2 points: TotalArray() correctly sums the relevant elements in the array and returns this sum.
3 points: AvgArray() correctly averages the relevant elements in the array and returns this average.
2 points: PrintArray() correctly prints the relevant elements in the array.
4 points: GetIntArray() works correctly and does not use the index [] operator.
4 points: TotalArray() works correctly and does not use the index [] operator.
4 points: AvgArray() works correctly and does not use the index [] operator.
4 points: PrintArray() works correctly and does not use the index [] operator.
4 points: Program is able to take a second array with a different count of elements.
4 points: Program prints the total of the two arrays together.
4 points: Program prints the average of the two arrays together.