CSC 215
Procedural Programming with C
Lab #6
Memory Allocation

Tutorial Section

- In the main method, do the following:
  - Include the library stdlib.h
  - Declare two integer pointers M and C. And integer size.
  - Read an integer from the keyboard and store it into size.
  - Allocate a dynamic memory for integers using **malloc** with size as its length. Give it to M.
  - Allocate a dynamic memory for integers using **calloc** with size as its length. Give it to C.
  - Print the content of array M and C. What's the difference?
  - Edit array M values with their indexes.
  - Print array M values.
  - Free the allocated memories.

```
=========================
Enter the size: 5
=========================
C[0] = 0        M[0] = 5911752
C[1] = 0        M[1] = 5903768
C[2] = 0        M[2] = 1128092492
C[3] = 0        M[3] = 1917869114
C[4] = 0        M[4] = 1634887535
=========================
After editing array M values with their indexes
M[0] = 0
M[1] = 1
M[2] = 2
M[3] = 3
M[4] = 4
=========================
```

Lab Section

- Write a program that does the following:
    - Ask the user to type the size of the array.
    - Use malloc or calloc to create an array of that size.
    - Use the function read to read the numbers.
    - Display the sum and average or these numbers. Then display the array sorted.
        - Show 2 numbers after the floating point in the average.
    - Free the allocated memory.

- Write the following functions:
    - Write the function ***read*** that takes a pointer to an integer (Array of integer) "Numbers" and an integer "size". Then read "size" number of integers and store them into "Numbers".
        - `void read(int *Numbers, int size)`
    - Write the function ***sum*** that takes a pointer to an integer (Array of integer) "Numbers" and an integer "size". Then return the sum of the integers in "Numbers".
        - `int sum(int *Numbers, int size)`
    - Write the function ***average*** that takes a pointer to an integer (Array of integer) "Numbers" and an integer "size". Then return the average of the integers in "Numbers".
        - `float average(int *Numbers, int size)`
    - Write the function ***sort*** that takes a pointer to an integer (Array of integer) "Numbers" and an integer "size". Then sort the integers in "Numbers" in an increasing order.
        - `void sort(int *Numbers, int size)`
    - Write the function ***read*** that takes a pointer to an integer (Array of integer) "Numbers" and an integer "size". Then prints the integers in "Numbers" separated by commas (,).
        - `void display(int *Numbers, int size)`

- Example runs:

```
$ ./lab6
How many numbers are you going to type? 5
Type 5 number(s): 32 -321 12 -4 23
The numbers sum     = -258
The numbers average = -51.60
The numbers sorted  : -321, -4, 12, 23, 32
```

- Show your program to the instructor. Then upload it to LMS under Lab6.

SUBMIT POLICY: -

- Use the follow naming convention: Lab06_ID_FirstName_LastName.c
    - **Example:** Lab06_123456789_Marwan_Almaymoni.c
- Use a comment to write your name and ID at the beginning of the code.
- The Deadline is: 06/04/2015 right before the Lab starts.
- Late submissions will not be accepted.
- Email submissions will not be accepted.
- **-1 Point** for not following the naming convention.
- **-1 Point** for not writing your name and ID in the code inside a comment.
- **-8 Points** if the submitted program didn't work due to syntax errors.
- **-10 Points** for cheating and helping others cheat.