**Programming assignment 1.**

**Suggested due date:** Friday, October 6 2017 at 3:30pm

Remember:

You can remove all the variables from the workspace by writing "clear"

**Look up the description of all the functions in MATLAB by typing doc in the command window.**

---

Implement *linearSearch(a,key)* and *binarySearch(a,key)* functions.

**Part A.** In this part we will calculate the **average-case running time** of each function.

1. Request the user to enter a positive integer, and call it **n**. ($n = 10^5$)
2. Generate **n** random integers between *-1000* to *1000* and save them in array **a**. *(You can use randi function in MATLAB)*
3. Sort **a** (you can use any sorting algorithm you want. If you are using MATLAB, you can call the *sort* function to sort your numbers*)*
4. Pick a random number in **a** and save it in variable called **key**.
5. Call each function separately to search for the key in the given array.
6. To calculate the **average-running time**, you need to have a timer to save the total runtime when repeating step 4 and 5 for **500** times. *(tic toc in MATLAB)*

(***Note1***: Do not forget to divide the runtime by the number of the times you run step 4-5)

(***Note2***: Remember to choose a different random number each time you go back to step 4.)

**Part B.** In this part we will calculate the **worst-case running time** of each function.

1. Repeat steps **1 to 3** in part A.
2. Now to have the worst-case scenario, set the value of the key to **5000** to make sure it does not exist in the array.
3. Run each function once to calculate the **worst-case running time** when $\mathbf{n = 10^5}$.
4. Calculate how much time your machine takes to run **one** single step. (***Hint***: look at HW3)
5. Now **estimate** the worst-case running time for each algorithm when $\mathbf{n=10^7}$. (***Hint***: look at HW3)
6. Now repeat step 1-3 for $\mathbf{n = 10^7}$ and compare the actual running time with your answer in step 5.