EEL3834 - Programming for Electrical Engineers Fall 2016
Programming Assignment 5
Assigned: 10/12/2016 Due: 10/19/2016 @ 4:00PM
To be done individually

In this homework, we will solve a problem using arrays. A palindrome is a word, phrase or number that reads the same forward and backward, ignoring blanks and considering uppercase and lowercase versions of the same letter to be equal. For example the following words/phrases in English are palindromes:

- "Eva can I stab bats in a cave"
- "Mr Owl ate my metal worm."
- "Was it a rat I saw",
- "A nut for a jar of tuna"
- "Ma is as selfless as I am"
- "Dammit, I'm mad"
- “Rats live on no evil star”
- "A Santa lived as a devil at NASA"
- Hannah
- Bob
- Ana
- Mike Kim

Palindromes are used in Computer Science, Mathematics, Acoustics and Biology. For example, recently it has been discovered that DNA palindromes appear frequently and are widespread in human cancers, and identifying them could help advance the understanding of genomic instability[1]. Your goal here is to write a program that will accept a sequence of characters and will decide whether the string is a palindrome. You may assume that **the input contains only letters and blanks** and is **at most 80 characters long**.

Output example **(Your code must match this exact format)**:

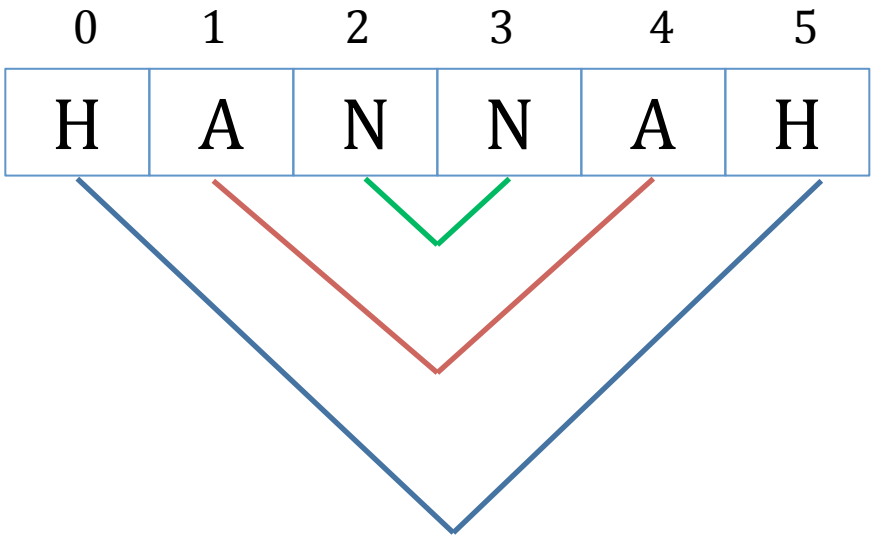
```
This program tests if a word/phrase is palindrome.

Please enter your phrase (just letters and blanks, please):

H a n n a h

Yes, the phrase is a palindrome!
```

[1] *Nature Genetics*

How to structure your program?

1) Your program should have a main method that asks the user for a new phrase. You should type each character of you phrase separated by a blank space character as in: "H a n n a h". You should store these characters (but not the blank spaces) in an array of characters.
2) Your program should invoke a method isPalindrome that receives as a parameter an array of characters and returns true if the set of characters in the array is a palindrome and false otherwise.
3) Your program returns to the user a message (e.g., "Yes, the phrase is a palindrome.") depending on the value returned by function isPalindrome.

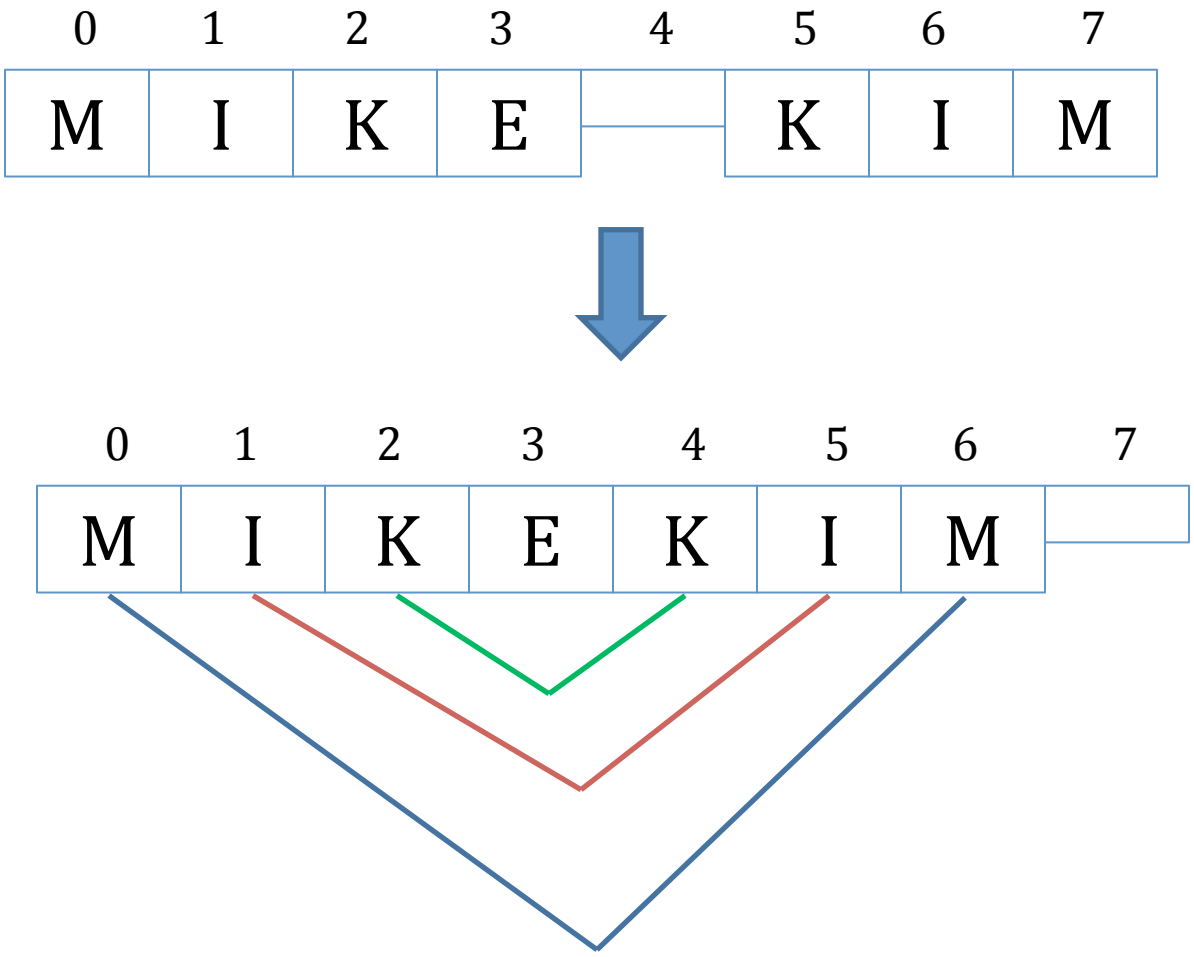
What does function isPalindrome do? Well, this function is the meat of your program and it will look at a string and decide whether or not it is palindrome. How to decide if a word is a palindrome? What would you do to discover if a phrase/word is a palindrome with just a piece of paper and a pencil:



Notice that in the example "Hannah" (6 characters) we start checking if the characters at the first position of the array (index 0) and at the character at the last position of the array (index 5) are the same. We then continue this process in a second iteration by comparing the character at first position + 1 (index 1) with character at last position – 1 (index 4). Then we continue this process in a third iteration by comparing the character at first position + 2 (index 2) and the character at the last position -2 (index 3).

When to stop? Have you noticed the number of times we repeated this process? Try this with pencil and paper for other words such as Abba, Anna, radar, etc… Did you discover any pattern?

Ok, but this is not enough as we may have a palindrome phrase and then we need to disregard blanks (white spaces first). What can we do? Again, what would you do if you had just a piece of paper and a pencil?



You will need a second array, where you will copy all the characters from the first arrays as long as they are not a white space.

You might find the function toupper() useful:
http://www.tutorialspoint.com/c_standard_library/c_function_toupper.htm

http://www.cplusplus.com/reference/cctype/toupper/

Programming style: Pay attention to issues of programming style:

- **use indentation**
- **comment your code/methods**
- **use meaningful names for variables**
- **leave spaces between logical blocks of the code**
- **comment your methods**

Call your source file ***assignment5***.cpp

Your grade will be subject to the following condition(s):

- Submission:
  The submission deadline is **4:00PM** on **10/19/16**. You will be penalized in increments of 25% per day late (regardless of the time). A submission at 4:01PM on 10/19/16 will result in a 25% penalty, as will a submission at 4:00PM on 10/20/16. A submission at 4:01PM on 10/20/16 will result in a 50% penalty, and so on. We will go by the timestamp on Canvas, so be sure to submit early.
  Submit your code on Canvas. You just need to **upload** your .cpp file, not copy and paste your code. In addition, you will need to write in the text entry box which version of g++ you used. This can be found from the terminal by typing: **g++ -v**.
  Also, PLEASE check your submission to make sure the file has actually been uploaded.

Your grade will be calculated based on the following (total 10 points)

- Compilation: **3 pts**
  Your code MUST compile in a Linux environment. Since that is the environment in which it will be graded. There is no partial credit available here, either your code compiles or it doesn't. Verify on a Linux terminal that your code compiles, runs, and behaves accordingly with the commands:

  g++ -o assignment5 assignment5.cpp
  ./assignment5

  (Review the commands in the setup guides from the beginning of the semester to know how to navigate the Linux Terminal)

- Execution: **4 pts**
  Your program will be tested against 8 cases, each worth 0.5 points. You can earn partial credit here if your code doesn't work for every single case. If it does work for every case, you will get the full 4 points. Note, for this assignment, the test cases will be slightly tougher, so make sure to account for all scenarios.

- Style: **3 pts**
  Your code will also be graded on its style. This includes things like using meaningful variable names, useful comments, proper indentation and spacing, and the use of functions. All of these things make your code easy to read and maintain. Partial credit will be available here. As a minimum, your code should have a comment at the beginning with your name, date, and a high level but still descriptive overview of what the program does.