CSE 2010, HW2, Fall 2018
Due Wed Sep 19 at the start of your lab section; Submit Server: class = cse2010, assignment = hw2S*x*Individual
Due Wed Sep 19 at the end of your lab section; Submit Server: class = cse2010, assignment = hw2S*x*GroupHelp
x is 2, 3, or 4—your section number or "j" for java submissions.

1 Written Part (30 points)

1. Explain the number of additons to the total (not Big-O) in terms of n for the following program segment:

```
int total = 0;
for (int i = 0; i < n; i += 2)
   total += i;    // addition to the total
```

2. Explain the number of additions to the total (not Big-O) in terms of n for the following program segment:

```
int total = 0;
for (int i = n; i > 0; i--)
   for (int j = 0; j < i; j++)
       total += i * j;     // addition to the total
```

3. Mathematically show that if $d(n)$ is $O(f(n))$ and $e(n)$ is $O(g(n))$, then the product $d(n)e(n)$ is $O(f(n)g(n))$

4. Consider $f(n) = 3n^2 + 2n - 1$, mathematically show that $f(n)$ is $O(n^2)$, $\Omega(n^2)$, and $\Theta(n^2)$.

5. For finding an item in a sorted array, consider "tertiary search," which is similar to binary search. It compares array elements at two locations and elminiates 2/3 of the array. To analyze the number of comparisons, the recurrence equations are $T(n) = 2 + T(n/3)$, $T(2) = 2$, and $T(1) = 1$, where n is the size of the array. Explain why the equations characterize "tertiary search" and solve for $T(n)$.

6. To analyze the time complexity of the "brute-force" algorithm in the programming part of this assignment, we would like to count the number of all possible multi-word phrases.

   (a) Explain the number of all possible phrases in terms of N (number of words in the input file) and L (length of a phrase; also known as *palLength* in the programming part).

   (b) Consider a computer that can process 1 billion phrases per second and N is 100, explain the number of years needed to process all possible phrases of length (L) 10.

   (c) If we don't want the computer to spend more than 1 minute, explain the largest N the computer can process phrases of length (L) 10.

2 Programming Part (70 points)

A palindrome is a string that reads the same forward and backward (excluding spaces and punctuations); for example, "wow" and "taco cat". Finding multi-word palindromes can yield interesting phrases.

The goal of the assignment is to find multi-word palindromes. Words in a multi-word palindrome are unique. The two main components of the problem are:

1. Design a recursive algorithm that checks if a string is a palindrome or not.

2. Given a list of words and *palLength* (number of words in a palindrome), design a recursive algorithm to find palindromes with *palLength* words.

Suggestion: Solve the first component (one recursive algorithm), before solving the second component (by adding a second recursive algorithm).

We will be evaluating your submission on code01.fit.edu; we recommend you to ensure that your submission runs on code01.fit.edu.

Input: Input is from the command-line arguments for hw2.c in this order:

1. filename of a list of words, one on each line

2. *palLength* (number of words in a palindrome), which is a positive integer

Output: The program prints palindromes in alphabetical/lexicographical order to the standard output (screen). Each palindrome is on a line.

Extra Credit (10 more points): Separate submission via hw2extra.c. Solve the ***entire*** problem without recursion (or using a stack to simulate recursion).

3 Submission

Submit hw2.c that has the main method and other program files. Submissions for Indvidual and GroupHelp have the same guidelines as HW1.

Submit the written part in PDF format to the Submit Server. Hardcopy is also acceptable in the lab. GroupHelp submission is not applicable to the written part.

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top.

For extra credit, submit hw2extra.c that has the main method and other program files. GroupHelp submission is not applicable to extra credit. Late submission for extra credit is not accepted.