CSE 2010, HW2
Due Tue Sep 24 at the start of your lab section; Submit Server: class = cse2010, assignment = hw2SxIndividual
Due Tue Sep 24 at the end of your lab section; Submit Server: class = cse2010, assignment = hw2SxGroupHelp
x is 2, 3—your section number (or j for java).

1 Written Part (30 points)

1. 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 = 0; i < n; i +=2)
    total += i;
```

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

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

4. Consider $f(n) = 5n^2 + 4n - 2$, 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 "ternary 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 strings.

   (a) Explain the number of all possible strings in terms of n (maximum length of a string).

   (b) Consider a computer that can process 1 billion strings per second and n is 100, explain the number of years needed to process all possible strings.

   (c) If we don't want the computer to spend more than 1 minute, explain the largest n the computer can process.

2 Programming Part (70 points)

Consider FBI has some suspects of cyber criminals and it would like to break into their online accounts to gather more evidence. Naturally, the accounts are password protected.

The goal of the assignment is to help FBI gain access to suspects' online accounts. Desgin a "brute-force" algorithm that **recursively** enumerates all possible strings. For simplicity, consider only lower-case letters and strings up to a certain maximum length. hw2_crypto.c contains the `encrypt(char[])` function that encrypts a string. Sample use of `encrypt(char[])` is in hw2.c. hw2.c, hw_crypto.c, sample input, and sample output are on the course website.

Input: The command-line argument for hw2.c is the name of the input file, which has:

1. the maximum length of a password on the first line
2. encrypted password on each following line

Output: Each line contains the password before encryption.

Extra Credit (10 more points): Separate submission via hw2_extra.c. Solve the 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 hw2_extra.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.