Introduction to Computer Science
HW #3
Due: 2014/04/23

## **Chapter 4 Review Problems** (8% each)**:**

Problems 7, 39, 40, 45, 48.

**M1.** (10%) Let's learn some html code. The following html produces the result in the right:

```
<!DOCTYPE html>
<html>
<body>

<table style="width:300px">
<tr>
   <td>Jill</td>
   <td>Smith</td>
   <td>50</td>
   </tr>
<tr>
   <td>Eve</td>
   <td>Jackson</td>
   <td>94</td>
</tr>
<tr>
   <td>John</td>
   <td>Doe</td>
   <td>80</td>
</tr>
</table>

</body>
</html>
```

| | | |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

Your task is to make the following changes: (1) table border be 1 , (2) Smith be bold, (3) Eve be red and size of 1, and (4) the cell of 80 be white font and black background. You need to post your **html code** and the **screenshot** of the result.

# Introduction to Computer Science
# HW #3
# Due: 2014/04/23

The following tags are useful (google them for more information):
<table border=….>
<font color=…. size=…>

<b>

## Programming Problem (50%):

**First, VERY IMPORTANT:** check if sizeof(unsigned long long int) or sizeof(unsigned long int) is 8. If not, use another computer.

Write two pieces of code:

(a) cipher.cpp reads the file "plain.txt" containing one string (length < 10000) and "public_key.txt" containing *N* and *e*. cipher.cpp should then output "secret.txt" as integers encrypted by RSA. The encoding concatenates 2 chars into one integer. For example, "AB" would be encoded as $(65*2^8+66)=16,706$. If only one char remains, put it to leftmost. For example, "A" would be encoded as $65*2^8=16,640$.

(b) decipher.cpp reads the file "secret.txt" and "private_key.txt" containing *N* and *d*. decipher.cpp should then output "message.txt" with content same as "plain.txt".

**Note:** Be careful about overflow, signed/unsigned, and eof() problem. "Ned.txt" contains more (*N*,*e*,*d*) sets for you to test.

## Bonus (5%)

Write the following function:

**unsigned long long int** findD(**unsigned long long int** e, **unsigned long long int** phi)

, which returns *d*, where $de \equiv 1 \pmod{phi}$. Save the function into bonus.cpp. No main(). Note: You need to use Euclidian algorithm. Enumeration won't earn any credit.