Problem Set #4—Inside the Machine

Due: Wednesday, October 16

Problem 1 (Chapter 6, review question 3, page 223)
Convert each of the following decimal numbers to its hexadecimal equivalent:

a. 17  
   c. 1729
b. 256  
   d. 2766

Problem 2 (Chapter 6, review question 4, page 223)
Convert each of the following hexadecimal numbers to decimal:

a. 17  
   c. CC
b. 64  
   d. FAD

Problem 3
In my lecture on functions on September 16, I wrote a program (which is in the ZIP file for that lecture on the web site) that found perfect numbers. Modify that program so that it also displays the binary form of these numbers. As you can see when you run the revised program, the first few perfect numbers follow an interesting pattern when you write them out in binary. Euclid discovered this pattern for the first four perfect numbers more than 2000 years ago, and the 18th-century Swiss mathematician Leonhard Euler proved that all even perfect numbers follow this pattern.

Problem 4 (Chapter 6, exercise 18, page 231)
Letter-substitution ciphers require the sender and receiver to use different keys: one to encrypt the message and one to decrypt it when it reaches its destination. Your task in this exercise is to write a function invertKey that takes an encryption key and returns the corresponding decryption key. In cryptography, that operation is called inverting the encryption key.

The idea of inverting a key is most easily illustrated by example. Suppose, for example, that the key is "QWERTYUIOPASDFGHJKLMNOPQRSTUVWXYZ" as in exercise 16. That key represents the following translation rule:

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ |
| Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M |

The translation table shows that A maps into Q, B maps into W, C maps into E, and so on. To turn the encryption process around, you have to read the translation table from bottom to top, looking to see what letter in the original text would have produced each letter in the encrypted version. For example, if you look for the letter A in the bottom line of the key, you discover that the corresponding letter in the original must have been K.
Similarly, the only way to get a B in the encrypted message is to start with an X in the original one. The first two entries in the inverted translation table therefore look like this:

```
 A  B
 V  X
```

If you continue this process by finding each letter of the alphabet on the bottom of the original translation table and then looking to see what letter appears on top, you will eventually complete the inverted table, as follows:

```
 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 V K X M C N O P H Q R S Z Y I J A D L E G W B U F T
```

The inverted key is simply the 26-character string on the bottom row, which in this case is "KXVMCNOPHQRSZYIJADLEGWBUFT".

Your solution to this problem will turn out to be useful on Project 4 (Enigma).

**Problem 5**

Using the material in Handout #17 as a reference, write a Toddler program called `Largest.asm` that allows the user to enter a list of integers and then reports the largest value. As in the `AddList.asm` program, the user indicates the end of the list by entering a zero value. The output of the program is illustrated by the following sample run:

```
<table>
<thead>
<tr>
<th>Toddler Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>? 314</td>
</tr>
<tr>
<td>? 159</td>
</tr>
<tr>
<td>? 265</td>
</tr>
<tr>
<td>? 358</td>
</tr>
<tr>
<td>? 979</td>
</tr>
<tr>
<td>? 323</td>
</tr>
<tr>
<td>? 846</td>
</tr>
<tr>
<td>? 0</td>
</tr>
<tr>
<td>979</td>
</tr>
</tbody>
</table>
```

You can use the Toddler simulator on the CSCI 121 web site to check your solution.