More Review Problems for CS112 Exam 2

Problem 1:
(a) Hand simulate the behavior of the mystery function (below) when called as follows:

```
>> num = mystery(14,3)
```

(b) What does the mystery function do?

(c) Rewrite the mystery function using a for loop instead of a while loop

*Note:* You cannot use any built-in division operator (e.g. / or rem). Using break is ok, but think about whether this can be done without using break.

```
function result = mystery (a, b)
    if (b > a)
        result = 0;
    else
        result = 0;
        val = a;
        while (val >= b)
            val = val - b;
            result = result + 1;
        end
    end
end
```

Problem 2:
The intent of the following code segment is to detect consecutive pairs of letters in a string that are identical (e.g. ‘ss’ and ‘pp’), and when such a pair is encountered, to remove the second letter. The final call to the disp function should print the string ‘misisipi’, but an error is encountered in the loop.

```
string = 'mississippi';
i = 1;
numLetters = length(string);
while (i < numLetters)
    if (string(i) == string(i+1))
        string(i+1) = '';
    end
    i = i + 1;
end
disp(string)
```

Note that in line 6, string(i+1) is assigned to an empty string, which removes the character from the string, as illustrated in the following example:

```
>> string = 'april';
>> string(3) = ''
string =
    apil
```
(a) What is the error that occurs, and what is the cause of this error? **Hint:** try to hand-simulate the code with a shorter string, e.g. suppose the first line of code is replaced with `string = 'eel'`.

(b) Make a modification to the code that fixes this error.

(c) Rewrite the code using a `for` statement. Assume that the variable `string` is assigned to ‘mississippi’ before the `for` statement, and that the statement `disp(string)` is placed after the `for` statement, and prints the string ‘misisipi’ when the code is executed.

*(See solutions on next two pages)*
Solutions to Problems

Problem 1:
(a) The following is a hand simulation of \( \text{num} = \text{mystery}(14,3) \)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>result</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>3</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

The value of \( \text{val} \) is now smaller than the value of \( b \), so the loop stops and the value of \( \text{result} \), which is 4 at the end, is returned and assigned to the variable \( \text{num} \).

(b) The \text{mystery} function returns the integer part of \( a/b \).

(c) The following is a compact approach to the \text{else} clause using a \text{for} loop:

```plaintext
else
    result = 0;
    for val = b\(\ldots\)a
        result = result + 1;
    end
end
```

Here are two approaches to the \text{else} clause with a \text{for} loop that is more similar to the approach used in the original \text{while} loop:

```plaintext
else
    val = a;
    for result = 1\(\ldots\)a
        val = val - b;
        if (val < b)
            break
        end
    end
end
else
    result = 0;
    val = a;
    for i = 1\(\ldots\)a
        val = val - b;
        result = result + 1;
        if (val < b)
            break
        end
    end
end
```
Problem 2:
(a) The string 'mississippi' has 11 characters, so numLetters is assigned to the value 11. As long as i is less than numLetters, i.e. i < 11, the while loop is entered. The problem is that when repeated letters are encountered, the second letter is removed, shortening the string. As a consequence, the reference to string(i+1) eventually generates an error, because the index i+1 is beyond the length of the string. The following hand-simulation illustrates the problem with the shorter string 'eel':

<table>
<thead>
<tr>
<th>string</th>
<th>i</th>
<th>numLetters</th>
<th>(i &lt; numLetters)</th>
<th>(string(i) == string(i+1))</th>
<th>(i+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>eel</td>
<td>1</td>
<td>3</td>
<td>true</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>el</td>
<td>2</td>
<td>3</td>
<td>true</td>
<td>error: i+1 is 3, but length of string is only 2</td>
<td></td>
</tr>
</tbody>
</table>

(b) The following code illustrates two ways to fix the error:

```matlab
string = 'mississippi';
i = 1;
while (i < length(string))
    if (string(i) == string(i+1))
        string(i+1) = ' ';  
        numLetters = length(string);  
        i = i + 1;
    end
end
disp(string)
```

(c) The following code implements the same task using a "for" statement, illustrating three different strategies:

```matlab
string = 'mississippi';
numLetters = length(string) - 1;
for i = 1:numLetters
    if (i >= length(string))
        break
    end
    if (string(i) == string(i+1))
        string(i+1) = ' ';  
        numLetters = length(string);  
        i = i + 1;
    end
end
disp(string)
```

```matlab
string = 'mississippi';
newString = string(1);
for i = 2:length(string)
    if (string(i) == string(i-1))
        newString = [newString string(i)];
    end
end
string = newString;
disp(string)
```