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
```

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 `num = mystery(14,3)`

a	b	result	val
14	3	0	14
		1	11
		2	8
		3	5
		4	2

The value of `val` is now smaller than the value of `b`, so the loop stops and the value of `result`, which is 4 at the end, is returned and assigned to the variable `num`.

(b) The `mystery` function returns the integer part of `a/b`.

(c) The following is a compact approach to the `else` clause using a `for` loop:

```
else
   result = 0;
   for val = b:b:a
       result = result + 1;
   end
end
```

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

```
else
   val = a;
   for result = 1:a
       val = val - b;
       if (val < b)
          break
       end
   end
end

else
   result = 0;
   val = a;
   for i = 1: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'`:

string	i	numLetters	(i < numLetters)	(string(i) == string(i+1))
eel	1	3	true	true
el	2	3	true	error: i+1 is 3, but length of string is only 2

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

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

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

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

```
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) = '';
    end
end
disp(string)
```

```
string = 'mississippi';
removals = [];
for i = 1:(length(string)-1)
   if (string(i) == string(i+1))
      removals = [removals i+1];
   end
end
string(removals) = '';
disp(string)
```

```
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)
```