2011Sp CS10 Online Final

Section 1

Instructions:
Save the file containing your answers with the name **`FinalFirstnameLastname.ypr`** (e.g., **`FinalBarackObama.ypr`**). You can assume that all the inputs that your program will ever be given are valid, so you do not have to perform any error checking. You will submit your solution on bSpace. When you are done, go to the `Assignments` tab and click on the `Online Final` assignment corresponding to your lab section. Upload your file, and *remember to click* `Submit`!

Take a deep breath and calm down before moving on: this exam is not worth having heart failure over … it's only worth 15 points or ~4% of your grade, about the same as one of the programming questions on the midterm. Good luck!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/sp11/exams/final/CS10-Starter.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

This block reports a list containing all of the numbers between 1 and 5, including start and ending values.

This block reports a list where each letter / number from the input word becomes a single element in the list.

This block reports the smaller of the two input numbers.

Question 1: We'll call a number a ***betty*** number if each of its digits is at least as large as the digit before it. This means that there are no digits that decrease in value as the number is read from left to right (so all single-digit numbers are ***betty*** numbers). Write a predicate that determines whether any positive number is a ***betty*** number; you may use any techniques you've learned in this class: recursion, higher-order functions, iteration, etc.

Some ***betty*** numbers: 14789, 45, 47899, 1122334, 3
Some non-***betty*** numbers: 12342, 9876, 152

Question 2: Write a block that reports a list containing all of the ***betty*** numbers in a given range. The user should be able to specify the minimum and maximum values of the range as parameters to your reporter block.

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 3: Write a block that takes a list of numbers and function as input, applies the function to each of the numbers in the list, and reports the **minimum value** out of all of the values produced as output by the function. Hopefully the table below helps explain what we mean.

Input function	Input list	Output
$f(x) = x^2$	2, 5, 9	4
f(x) = x mod 10	17, 25, 43	3
$f(x) = (x-6)^2$	1, 5, 9	1

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 4: Using the blocks above and anything else you might need, have your sprite say (when the green flag is clicked) the minimum value of the function:

```
f(n) = (1021 * n) mod 101
```

...for all ***betty*** numbers between 1 and 100. *Hint: your sprite should say 1.*

2011Sp CS10 Online Final

Section 2

Instructions:
Save the file containing your answers with the name **`FinalFirstnameLastname.ypr`** (e.g., **`FinalBarackObama.ypr`**). You can assume that all the inputs that your program will ever be given are valid, so you do not have to perform any error checking. You will submit your solution on bSpace. When you are done, go to the `Assignments` tab and click on the `Online Final` assignment corresponding to your lab section. Upload your file, and *remember to click* `Submit`!

Take a deep breath and calm down before moving on: this exam is not worth having heart failure over … it's only worth 15 points or ~4% of your grade, about the same as one of the programming questions on the midterm. Good luck!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/sp11/exams/final/CS10-Starter.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

This block reports a list containing all of the numbers between 1 and 5, including start and ending values.

This block reports a list where each letter / number from the input word becomes a single element in the list.

This block reports the larger of two input numbers.

Question 1: We'll call a number a ***ted*** number if each of its digits is at most as large as the digit before it. This means that there are no digits that increase in value as the number is read from left to right (so all single-digit numbers are ***ted*** numbers). Write a predicate that determines whether any positive number is a ***ted*** number; you may use any techniques you've learned in this class: recursion, higher-order functions, iteration, etc.

Some ***ted*** numbers: 3, 75441, 851, 41
Some non-***ted*** numbers: 152, 712, 145

Question 2: Write a block that reports a list containing all of the ***ted*** numbers in a given range. The user should be able to specify the minimum and maximum values of the range as parameters to your reporter block.

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 3: Write a block that takes a list of numbers and function as input, applies the function to each of the numbers in the list, and reports the **maximum value** out of all of the values produced as output by the function. Hopefully the table below helps explain what we mean.

Input function	Input list	Output
$f(x) = x^2$	2, 5, 9	81
f(x) = x mod 10	17, 25, 43	7
$f(x) = (x-6)^2$	1, 5, 9	25

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 4: Using the blocks above and anything else you might need, have your sprite say (when the green flag is clicked) the maximum value of the function:

```
f(n) = (1091 * n) mod 89
```

...for all ***ted*** numbers between 1 and 100. *Hint: your sprite should say 86.*

2011Sp CS10 Online Final

Section 3

Instructions:
Save the file containing your answers with the name **`FinalFirstnameLastname.ypr`** (e.g., **`FinalBarackObama.ypr`**). You can assume that all the inputs that your program will ever be given are valid, so you do not have to perform any error checking. You will submit your solution on bSpace. When you are done, go to the `Assignments` tab and click on the `Online Final` assignment corresponding to your lab section. Upload your file, and *remember to click* `Submit`!

Take a deep breath and calm down before moving on: this exam is not worth having heart failure over … it's only worth 15 points or ~4% of your grade, about the same as one of the programming questions on the midterm. Good luck!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/sp11/exams/final/CS10-Starter.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

This block reports a list containing all of the numbers between 1 and 5, including start and ending values.

This block reports a list where each letter / number from the input word becomes a single element in the list.

Question 1: We'll call a number a ***volatile*** number if its digits alternate between being higher and lower than the previous digit. All single-digit numbers are ***volatile*** numbers. Write a predicate that determines whether any positive number is a ***volatile*** number; you've may use any techniques you learned in this class: recursion, higher-order functions, iteration, etc.

Some ***volatile*** numbers: 3, 471, 4917, 152
Some non-**volatile** numbers: 213, 789, 432, 1221

Question 2: Write a block that reports a list containing all of the ***volatile*** numbers in a given range. The user should be able to specify the minimum and maximum values of the range as parameters to your reporter block.

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 3: Write a block that takes a list of numbers and function as input, applies the function to each of the numbers in the list, and reports the **sum** of all of the values produced as output by the function. Hopefully the table below helps explain what we mean.

Input function	Input list	Output
$f(x) = x^2$	2, 5, 9	110
f(x) = x mod 10	17, 25, 43	15
$f(x) = (x-6)^2$	1, 5, 9	35

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 4: Using the blocks above and anything else you might need, have your sprite say (when the green flag is clicked) the sum of the outputs of the function:

$$f(n) = n^2$$

...for all ***volatile*** numbers between 1 and 100. *Hint: your sprite should say 73425.*

2011Sp CS10 Online Final

Section 4

Instructions:
Save the file containing your answers with the name **`FinalFirstnameLastname.ypr`** (e.g., **`FinalBarackObama.ypr`**). You can assume that all the inputs that your program will ever be given are valid, so you do not have to perform any error checking. You will submit your solution on bSpace. When you are done, go to the `Assignments` tab and click on the `Online Final` assignment corresponding to your lab section. Upload your file, and *remember to click* `Submit`!

Take a deep breath and calm down before moving on: this exam is not worth having heart failure over … it's only worth 15 points or ~4% of your grade, about the same as one of the programming questions on the midterm. Good luck!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/sp11/exams/final/CS10-Starter.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

This block reports a list containing all of the numbers between 1 and 5, including start and ending values.

This block reports a list where each letter / number from the input word becomes a single element in the list.

Question 1: We'll call a number a ***differing*** number if the each digit differs by at least one from digit before it. All single-digit numbers are ***differing*** numbers. Write a predicate that determines whether any positive number is a ***differing*** number; you may use any techniques you've learned in this class: recursion, higher-order functions, iteration, etc.

Some ***differing*** numbers: 3, 184, 485, 131, 147
Some non-***differing*** numbers: 231, 12, 711

Question 2: Write a block that reports a list containing all of the ***differing*** numbers in a given range. The user should be able to specify the minimum and maximum values of the range as parameters to your reporter block.

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 3: Write a block that takes a list of numbers and function as input, applies the function to each of the numbers in the list, and reports the **product** of all of the values produced as output by the function. Hopefully the table below helps explain what we mean.

Input function	Input list	Output
$f(x) = x^2$	2, 1, 4	64
f(x) = x mod 10	17, 25, 43	105
$f(x) = (x-6)^2$	1, 5, 6	0

You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from ToolSprite. You will find the blocks that you created in earlier questions useful.

Question 4: Using the blocks above and anything else you might need, have your sprite say (when the green flag is clicked) the product of the outputs of the function:

```
f(n) = sqrt(n)
```

...for all ***differing*** numbers between 1 and 100.

Hint: your sprite should say 3.884430620002423e56