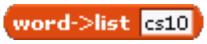
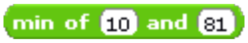
2011Fa CS10 Online Final

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`!

Preparation:
Go to **http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

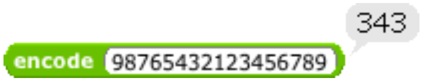
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 min of 10 and 81	This block reports the smaller of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, etc.

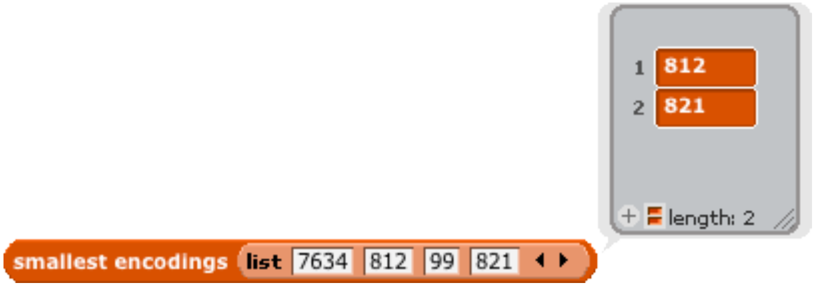
So, for example, if your input number is 98765432123456789, then  encode 98765432123456789 343 I.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
  9   8   7   6   5   4   3   2   1   2   3   4   5   6   7   8   9 ← your number
* 1   2   2   3   3   3   4   4   4   4   5   5   5   5   5   6   6 ← coefficients
-------------------------------------------------------------------
  9 +16 +14 +18 +15 +12 +12  +8  +4  +8 +15 +20 +25 +30 +35 +48 +54 = 343 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a **`smallest-encodings`** block that takes a list of numbers and returns **a list of these numbers that have the smallest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers **`812`** and **`821`** have the smallest encodings of all the numbers in the list **`(7634 812 99 821)`**, so our function returns a list of these two elements **`(812 821)`**. If only *one* number is the smallest, the function should return a list of that one element.

Original Number	7634	812	99	821
Encoded number	7*1+6*2+3*2+4*3=**37**	8*1+1*2+2*2 = **14**	9*1+9*2 = **27**	8*1+2*2+1*2 = **14**



smallest encodings list 7634 812 99 821

1 812
2 821
length: 2

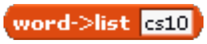
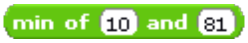
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa 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`!

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

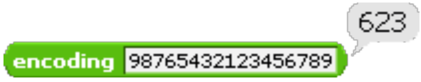
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 min of 10 and 81	This block reports the smaller of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

1, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 8, 8, 16, 16, 16, etc.

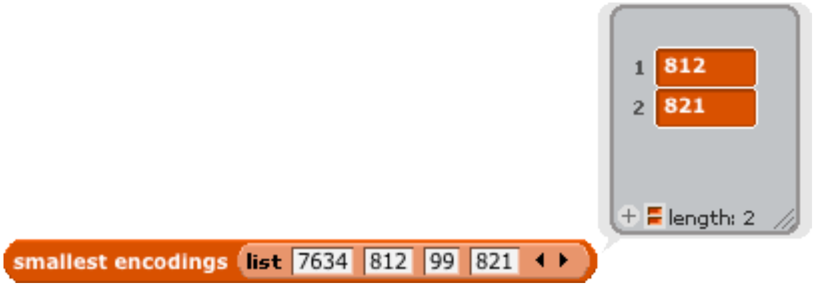
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 623 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
   9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9 ← your number
*  1    2    2    4    4    4    4    8    8    8    8    8    8    8    8   16   16 ← coefficients
-------------------------------------------------------------------------------------
   9  +16  +14  +24  +20  +16  +12  +16   +8  +16  +24  +32  +40  +48  +56+128+144 = 623 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a **`smallest-encodings`** block that takes a list of numbers and returns **a list of these numbers that have the smallest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers **`812`** and **`821`** have the smallest encodings of all the numbers in the list **`(7634 812 99 821)`**, so our function returns a list of these two elements **`(812 821)`**. If only *one* number is the smallest, the function should return a list of that one element.

Original Number	7634	812	99	821
Encoded number	7*1+6*2+3*2+4*4=**41**	8*1+1*2+2*2 = **14**	9*1+9*2 = **27**	8*1+2*2+1*2 = **14**



```
1  812
2  821
length: 2

smallest encodings  list 7634 812 99 821
```

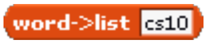
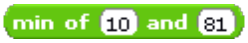
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa 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`!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

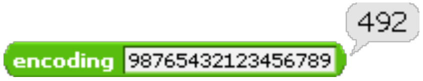
Block	**Description**
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 min of 10 and 81	This block reports the smaller of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

2, 2, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, 8, 8, 8, 8, 8, 8, etc.

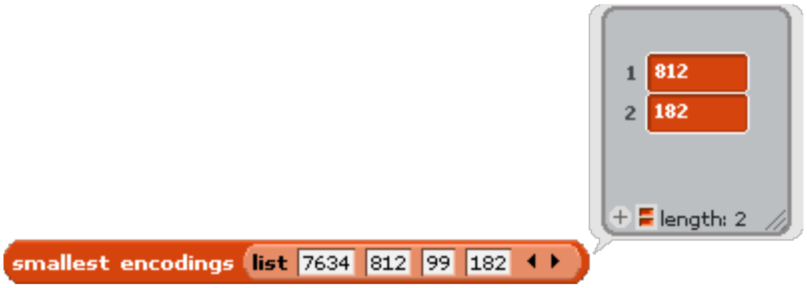
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 492 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
  9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9 ← your number
* 2    2    4    4    4    4    6    6    6    6    6    6    8    8    8    8    8 ← coefficients
--------------------------------------------------------------------
 18 +16  +28  +24  +20  +16  +18  +12   +6  +12  +18  +24  +40  +48  +56  +64  +72 = 492 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a `smallest-encodings` block that takes a list of numbers and returns **a list of these numbers that have the smallest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers **`812`** and **`182`** have the smallest encodings of all the numbers in the list **`(7634 812 99 182)`**, so our function returns a list of these two elements **`(812 182)`**. If only *one* number is the smallest, the function should return a list of that one element.

Original Number	7634	812	99	182
Encoded number	7*2+6*2+3*4+4*4=**54**	8*2+1*2+2*4 = **26**	9*2+9*2 = **36**	1*2+8*2+2*4 = **26**



1 812
2 182
length: 2

smallest encodings list 7634 812 99 182

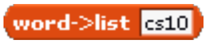
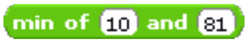
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa 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`!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

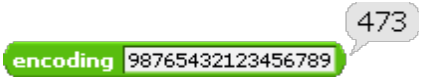
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 min of 10 and 81	This block reports the smaller of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

1, 3, 3, 3, 5, 5, 5, 5, 5, 7, 7, 7, 7, 7, 7, 7, 9, 9, etc.

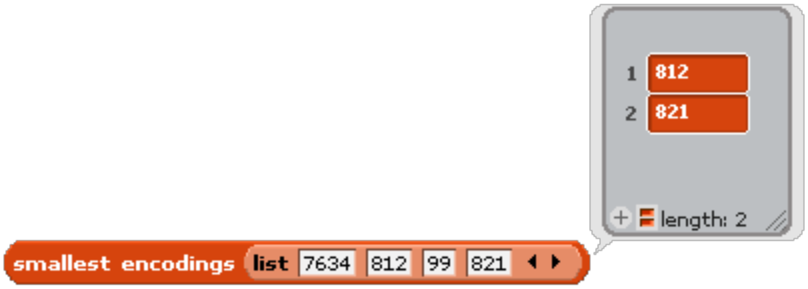
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 473 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
  9   8   7   6   5   4   3   2   1   2   3   4   5   6   7   8   9 ← your number
* 1   3   3   3   5   5   5   5   5   7   7   7   7   7   7   7   9 ← coefficients
--------------------------------------------------------------------
  9 +24 +21 +18 +25 +20 +15 +10  +5 +14 +21 +28 +35 +42 +49 +56 +81 = 473 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a `smallest-encodings` block that takes a list of numbers and returns **a list of these numbers that have the smallest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers `812` and `182` have the smallest encodings of all the numbers in the list `(7634 812 99 182)`, so our function returns a list of these two elements `(812 821)`. If only *one* number is the smallest, the function should return a list of that one element.

Original Number	7634	812	99	821
Encoded number	7*1+6*3+3*3+4*3=**46**	8*1+1*3+2*3 = **17**	9*1+9*3 = **36**	8*1+2*3+1*3 = **17**



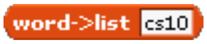
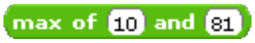
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa CS10 Online Final

Section 5

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`!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

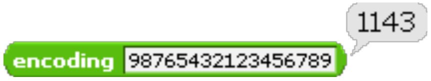
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 max of 10 and 81	This block reports the larger of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

1, 3, 3, 3, 9, 9, 9, 9, 9, 9, 9, 9, 9, 27, 27, 27, 27, 27, 27, etc.

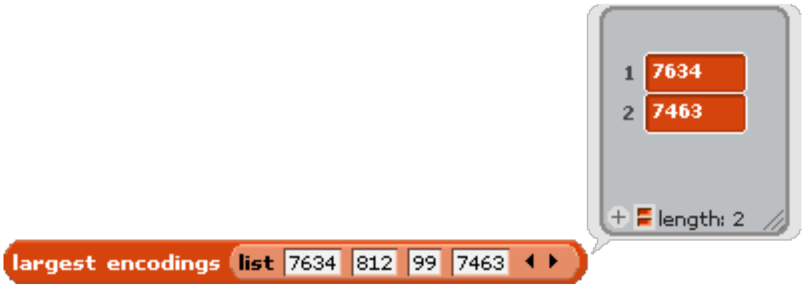
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 1143 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
  9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9 ← your number
* 1    3    3    3    9    9    9    9    9    9    9    9    9   27   27   27   27 ← coefficients
--------------------------------------------------------------------
  9  +24  +21  +18  +45  +36  +27  +18   +9  +18  +27  +36  +45+162+189+216+243 = 1143 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a `largest-encodings` block that takes a list of numbers and returns **a list of these numbers that have the largest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers **`7634`** and **`7463`** have the largest encodings of all the numbers in the list **`(7634 812 99 7463)`**, so our function returns a list of these two elements **`(7634 7463)`**. If only *one* number is the largest, the function should return a list of that one element.

Original Number	7634	812	99	7463
Encoded number	7*1+6*3+3*3+4*3=**46**	8*1+1*3+2*3 = **17**	9*1+9*3 = **36**	7*1+4*3+6*3+3*3=**46**



largest encodings list 7634 812 99 7463

1 7634
2 7463
length: 2

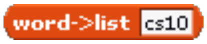
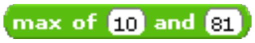
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa CS10 Online Final

Section 6

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`!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

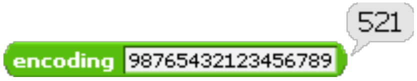
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 max of 10 and 81	This block reports the larger of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

3, 3, 3, 5, 5, 5, 5, 5, 7, 7, 7, 7, 7, 7, 7, 9, 9, 9, etc.

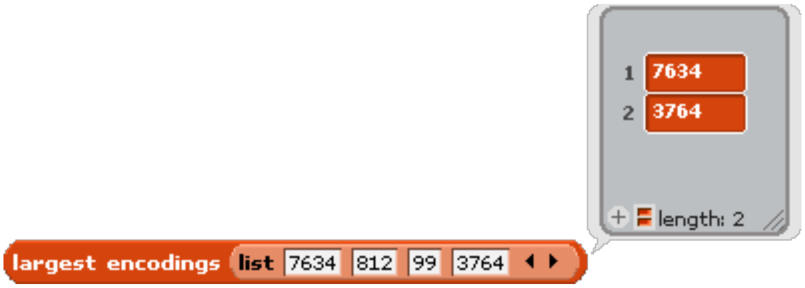
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 521 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
   9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9 ← your number
*  3    3    3    5    5    5    5    5    7    7    7    7    7    7    7    9    9 ← coefficients
--------------------------------------------------------------------
  27  +24  +21  +30  +25  +20  +15  +10   +7  +14  +21  +28  +35  +42  +49  +72  +81  =  521 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a `largest-encodings` block that takes a list of numbers and returns **a list of these numbers that have the largest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers **`7634`** and **`3764`** have the largest encodings of all the numbers in the list **`(7634 812 99 3764)`**, so our function returns a list of these two elements **`(7634 3764)`**. If only *one* number is the largest, the function should return a list of that one element.

Original Number	7634	812	99	3764
Encoded number	7*3+6*3+3*3+4*5=**68**	8*3+1*3+2*3 = **33**	9*3+9*3 = **36**	3*3+7*3+6*3+4*5=**68**



```
largest encodings (list 7634 812 99 3764)
```

```
1 7634
2 3764
length: 2
```

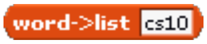
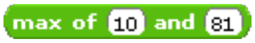
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa CS10 Online Final

Section 7

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`!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

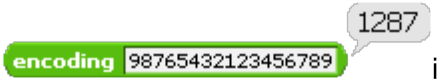
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 max of 10 and 81	This block reports the larger of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

3, 3, 3, 9, 9, 9, 9, 9, 9, 9, 9, 9, 27, 27, 27, 27, 27, 27, etc.

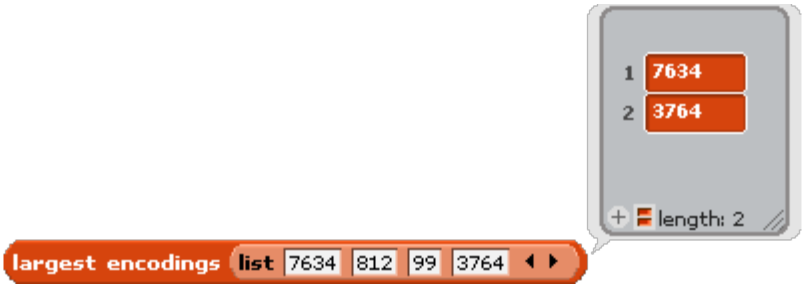
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 1287 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
  9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9 ← your number
* 3    3    3    9    9    9    9    9    9    9    9    9   27   27   27   27   27 ← coefficients
-------------------------------------------------------------------------------------
 27  +24  +21  +54  +45  +36  +27  +18   +9  +18  +27  +36+135+162+189+216+243 = 1287 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a `largest-encodings` block that takes a list of numbers and returns **a list of these numbers that have the largest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers `7634` and `3764` have the largest encodings of all the numbers in the list `(7634 812 99 3764)`, so our function returns a list of these two elements `(7634 3764)`. If only *one* number is the largest, the function should return a list of that one element.

Original Number	7634	812	99	3764
Encoded number	7*3+6*3+3*3+4*9=**84**	8*3+1*3+2*3 = **33**	9*3+9*3 = **54**	3*3+7*3+6*3+4*9=**84**



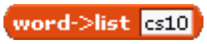
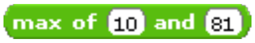
You should use higher-order functions in your solution. You may not use explicit recursion or the yellow and orange loop constructs (such as `repeat`, `repeat until`, and `for`); you can, however, use the built-in list blocks in the `Variables` tab and the HOF blocks that have already been imported from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

2011Fa CS10 Online Final

Section 8

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`!

Preparation:
Go to **`http://inst.eecs.berkeley.edu/~cs10/fa11/exams/final/CS10-Base.ypr`** and download the starter project. This will contain the blocks from *ToolSprite* as well as two blocks that we've built for you:

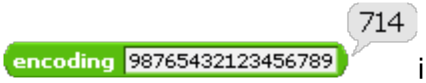
Block	Description
 word->list cs10	This block reports a list where each letter / number from the input word becomes a single element in the list.
 max of 10 and 81	This block reports the larger of the two input numbers.

Partial Credit: This exam has two questions, and you can get full credit for one of them even if you haven't written the other one. So, if you get stuck on question 1, move on to question 2.

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

Question 1: Write an **`encode`** function that takes a number, and returns an *encoded* number. To encode a number, you multiply each of the digits (from left to right) of the original number by a particular coefficient, and sum the result. The coefficients are taken from the infinite series:

2, 2, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 8, 8, 16, 16, 16, 16, etc.

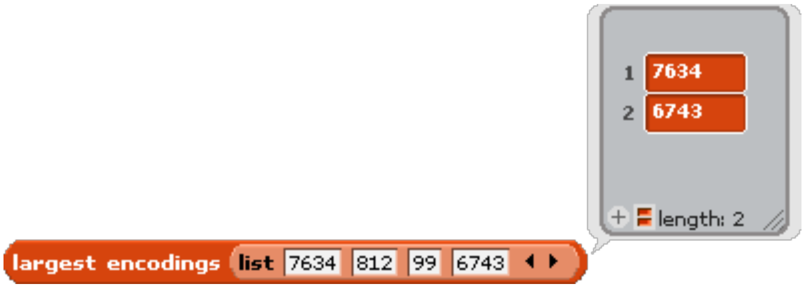
So, for example, if your input number is 98765432123456789, then  encoding 98765432123456789 714 i.e., to encode we multiply each of the input digits with the coefficients by lining them up as shown, multiplying the columns, and summing the result, as in the calculation below.

```
   9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9 ← your number
 * 2    2    4    4    4    4    8    8    8    8    8    8    8    8   16   16   16 ← coefficients
-----------------------------------------------------------------
  18  +16  +28  +24  +20  +16  +24  +16   +8  +16  +24  +32  +40  +48+112+128+144  = 714 ← encoding
```

You may use any techniques you've learned in this class for Question 1: recursion, higher-order functions, iteration, etc. *You have to procedurally generate the coefficients* -- you can't just type them in. That is, you have to be able to encode numbers that are millions of digits long.

Question 2: Write a `largest-encodings` block that takes a list of numbers and returns **a list of these numbers that have the largest encodings**. *We don't want the encoded numbers, we want the original numbers.* So, for example, the numbers **`7634`** and **`6743`** have the largest encodings of all the numbers in the list **`(7634 812 99 6743)`**, so our function returns a list of these two elements **`(7634 6743)`**. If only *one* number is the largest, the function should return a list of that one element.

Original Number	7634	812	99	6743
Encoded number	7*2+6*2+3*4+4*4=**54**	8*2+1*2+2*4 = **26**	9*2+9*2 = **36**	6*2+7*2+4*4+3*4=**54**



largest encodings list 7634 812 99 6743

1 7634
2 6743
length: 2

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.