

CS10 Quest- Summer 2019

Your Name (first, last)

ID Card Number

Your TA's Name

← _____
Name of person on left (or aisle)

_____ →
Name of person on right (or aisle)

Fill in the correct circles & squares completely...like this: ● (select ONE) ■ (select ALL that apply)

Questions 1-2: What's that Smell? It's Potpourri! (2 points each, 4 points total)

Question 1: Which of the following is **not** an example of Abstraction?

- The pedal that makes a car move is called the “accelerator” instead of “gas pedal”, because we don't need to know if the car actually uses gas.
- Rather than giving different directions of how to feed each animal, we can just give a single direction of how to feed any type of animal.
- We want our map of the train station to have as many details as possible to ensure we don't get lost.
- We can use a smartphone without understanding the code and hardware that make it work.
- All of the above are examples of Abstraction.

Question 2: We are writing a **function** to report the *absolute value* of a number. We test it with the input “-10” and our **function** correctly reports “10” as the *absolute value*. From this, we know that ...

- Our function is correct.
- Our function works correctly for any number input, but we don't know what it will report for text inputs.
- Our function works correctly for any negative number input, but we still don't know if it works for positive numbers.
- Our function works correctly for -10, but we still don't know if it works for other inputs.
- We can't determine any of the above.

Question 3: Number Conversions (4 points)

What is $12_{10} + 121_3$ in hexadecimal? Write your answer in the box to the right.

0x1C

Explanation: We first convert 121_3 to decimal so it's in the same base as our other number: $121_3 = 1 \times 3^0 + 2 \times 3^1 + 1 \times 3^2 = 16_{10}$. Now, we add this to 12_{10} , getting 28_{10} . Since it's difficult to convert directly from decimal to hex, we then convert 28_{10} to binary, giving us 11100_2 . Then, we use our chart to convert this to hex, which gives us $1C_{16}$.

Question 4: Ands and Ors Galore (5 + 2 = 7 points)

For this problem, consider the block below:



- a. Assuming this expression has no errors, what do we **definitely** know about each of the variables and blocks above? You may select multiple choices per row and column.

	Numbers	Text	Booleans	Lists	Not enough information
Data type of x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Data type of y	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Domain of bar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Range of foo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Range of bar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Explanation:

- Data type of **x**: All we know about **x** is that it's passed into **foo**. We don't know anything else about **x** or the domain of **foo**, so there isn't enough information to tell what the data type of **x** is.
- Data type of **y**: Since **y** is an input to the **+** block, we definitely know that the data type of **y** includes numbers.
- Domain of **bar**: Since **bar** takes in **y** as input, the domain of **bar** includes the data type of **y**, which is numbers.
- Range of **foo**: Since the output of **foo** is passed into the **item** block, which takes in lists, **foo**'s range must include lists.
- Range of **bar**: Since the output of **bar** is passed into the **+** block, which takes in numbers, **bar**'s range must include numbers.

b. Now, assume that the whole expression outputs true. Which of the following do we definitely know?

-  must report true
-  must report false
-  must report true
-  must report false
- It is impossible for the expression to report true (do not select any of the previous answers if you select this one)
- None of the above

Explanation: First, we can analyze the following expression from our code:

. Since there is an or true in this expression, it must output true, regardless of what the first part of the expression outputs. Then, we notice that this

expression is actually wrapped with a not:

. Since the expression inside the not always reports true, with the not outside it, it always reports false. Now, we can reduce our expression above to

. Because of how and behaves, this expression will always report false.

Question 5: Lisztomania (2 points each, 8 points total)

In the box to the right of each script below, write down what the sprite would say after running. You may assume that each script is run independently.

To write down a list, separate each item with commas. For example,  can be written as 1, 2, 3.

- a. Explanation: There is a very similar example/explanation on the lecture slides [here](#).

```

script variables x y
set x to list 1 2 3
set y to x
delete 1 of x
say x for 2 secs
  
```

- b. Explanation: There is a very similar example/explanation on the lecture slides [here](#).

```

script variables x y
set x to list 1 2 3
set y to x
delete 1 of x
say y for 2 secs
  
```

- c. Explanation: There is a very similar example/explanation on the lecture slides [here](#)

```

script variables x y
set x to 1
set y to x
change y by 1
say x for 2 secs
  
```

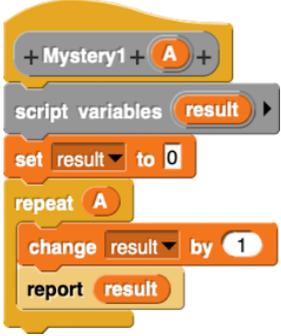
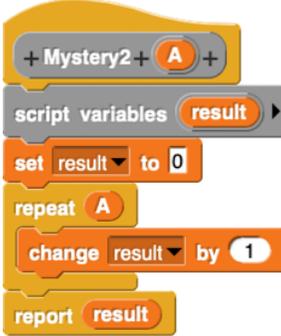
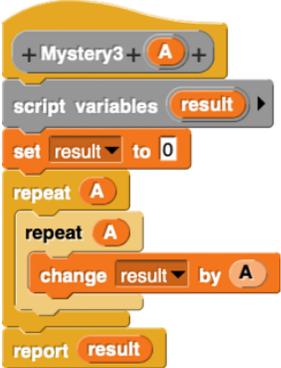
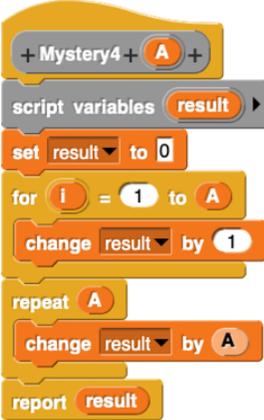
- d. Explanation: There is a very similar example/explanation on the lecture slides [here](#).

```

script variables x y
set x to list 1 2 3
set y to x
set y to list a b c
say x for 2 secs
  
```

Question 6: The Magical Mystery Machine (2 points each, 16 points total)

For each of the following functions, indicate both what it reports and what the running time is.

			
---	---	--	---

What does each function report? You may assume A is a positive integer. Choose one option for each row below.

	1	A	$2A$	A^2	$A^2 + A$	A^3	2^A
Mystery1 A	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mystery2 A	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mystery3 A	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Mystery4 A	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Explanation:

- **Mystery1:** This function reports immediately after changing `result` by 1 once.
- **Mystery2:** This function changes `result` by 1 A times.
- **Mystery3:** This function has two `repeat A` loops, one inside of the other. The inside loop changes `result` by A A times, so it changes `result` by A^2 each time it runs. The outside loop makes the inside loop run A times, meaning `result` changes by $A^2 A$ times, making `result`'s final value A^3 .
- **Mystery4:** This function first changes `result` by 1 A times, making `result`'s value after the loop A . Then, the second loop changes `result` by A A times, adding A^2 to `result`, for a final value of $A^2 + A$.

What is the running time of each function? Choose one option for each row below.

	Constant	Logarithmic	Linear	Quadratic	Exponential
Mystery1 A	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mystery2 A	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mystery3 A	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Mystery4 A	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Explanation:

- **Mystery1:** This function reports immediately after changing `result` by 1 once, so no matter how big the input is, this function will report immediately after one iteration of the loop.
- **Mystery2:** This function changes `result` by 1 A times, meaning there is a total of A work.
- **Mystery3:** The inner loop repeats a constant time operation A times, meaning it takes A time. The loop outside of it repeats this process A times, meaning the total runtime is $A \times A = A^2$.
- **Mystery4:** This runs two loops, each of which takes A times sequentially, leading to $2A$ time, which is linear.

Question 7: An Expansion of Expand (6 points)

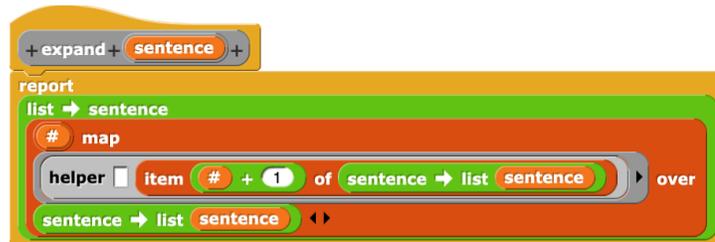
Recall the **expand** block from lab, which takes in a sentence and changes it based on the following rule: if a word is directly after a number, the word is repeated that number of times and the number does not appear in the output. Here are some example calls to **expand**:



We'd like to rewrite the **expand** block, this time using a special version of **map**. This version of **map** has an additional variable, **#**, which refers to the index of the element. Here are some example calls to this special **map**:



Here is our new **expand** block, written with the special **map**:



Write out the code for **helper** in the box below to make **expand** work properly. Feel free to use the following block in your code, which takes in an element and repeats it a certain number of times:



Hint: Since **#** represents the index of the current element, **# + 1** will represent the index of the next element. If the current element is the last element of the list, **# + 1** will be the index of an element that doesn't exist! For this problem, you do not need to worry about this and can trust that the given code does not error.

Implement the function **helper** in the box below so that **expand** works as explained.

helper (input1) (input2):



Question 8: Pair-Mutations (6 + 6 = 12 points)

We are trying to write a command block called **swap pairs** that will have two lists as input. Both lists will contain numbers, and both lists **must** have the same length. Each **pair** is the two items at the same index from the two lists. (eg. item 1 of list **A** and item 1 of list **B** form a **pair**, item 2 of list **A** and item 2 of list **B** form a different **pair**, etc.) For every pair in our lists, we want this block to put the smaller value into list **A** and the larger value into list **B**. If both items in the pair have the same value, then the values in that pair remain the same.

For example, after running the following code:



list **A** should have the values:

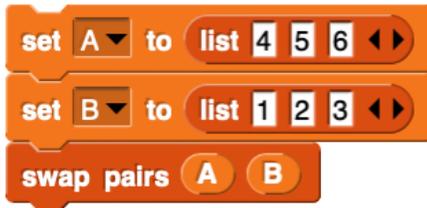


list **B** should have the values:



(No values were changed because for each **pair**, the smaller value was already in list **A**)

After running this code:



list **A** should have the values:



list **B** should have the values:



(The values are switched in every **pair** because $4 > 1$, $5 > 2$, $6 > 3$.)

After running this code:



list **A** should have the values:



list **B** should have the values:



(For the **pair** at the first index, $1 < 2$ so the values stay the same. For the **pair** at the second index, $5 = 5$ so the values stay 5. For the **pair** at the third index, $10 > 8$ so the values are switched.)

- a. We started writing the block, but it doesn't always work correctly! To help us debug our code, indicate what the resulting values will be for list **A** and list **B** using our **incorrect** block. Here is our block:

```

+ swap + pairs + (buggy) + list1 + list2 +
for i = 1 to length of list1
if item i of list1 > item i of list2
  replace item i of list1 with item i of list2
  replace item i of list2 with item i of list1

```

What will be the values of **A** and **B** after running the following 3 scripts? Write your answers directly into the empty slots of the *list* blocks.

	A	B
<pre> set A to list 1 2 3 set B to list 4 5 6 swap pairs (buggy) A B </pre>	<pre>list 1 2 3</pre>	<pre>list 4 5 6</pre>
<pre> set A to list 4 5 6 set B to list 1 2 3 swap pairs (buggy) A B </pre>	<pre>list 1 2 3</pre>	<pre>list 1 2 3</pre>
<pre> set A to list 1 5 10 set B to list 2 5 8 swap pairs (buggy) A B </pre>	<pre>list 1 5 8</pre>	<pre>list 2 5 8</pre>

- b. Now that we know how our buggy solution behaves, it's time for you to help us create a working block! On the lines below, write out code to implement a working version of swap pairs. Just like in the original code, your block must mutate the original lists: it may not product new lists. You may only use the lines given, though you may not need to use all of them. Please write clearly!

swap pairs (list1) (list2):

