Given below are the condition possibilities for an if statement

Below are the tiles at the bottom of a **procedure**

Below are the tiles at the bottom of a **function**
Given below are the panda procedures and panda Properties on the bottom right.
Given below are the panda functions.
If, loops, and changing an array element.

Random Integer and Decimal Numbers

<table>
<thead>
<tr>
<th>Random</th>
<th>nextRandomIntegerFrom0UpToButExcludingN</th>
<th>nextRandomIntegerFromAUpToButExcludingB</th>
<th>nextRandomIntegerFromAUpToAndIncludingB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal to Whole Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Math</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom WholeNumber...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Random</th>
<th>nextRandomRealNumberInRange [0.0, 1.0]</th>
<th>nextRandomRealNumberInRange [???, ???]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole to Decimal Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Math</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom DecimalNumber...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Events

```plaintext
declare procedure sceneActivated
do in order
  this myFirstMethod

declare procedure timeElapsed
do in order
  drop statement here

declare procedure keyPressed
  event isLetter
  event isDigit
  event getKey
  event isKey
  key:
  if (event isKey) is true then
    drop statement here
  else
    drop statement here

declare procedure mouseClicked
  event getScreenDistanceFromLeft
  event getScreenDistanceFromBottom
  event getModelAtMouseLocation
  if (event getModelAtMouseLocation == (this.panda)) is true then
    drop statement here
  else
    drop statement here

declare procedure collisionStarted
  event getSThingFromSetA
  event getSThingFromSetB
  drop statement here
```
1. (10 pts) **PART A:** Consider the following snapshot of an Alice world in which there are three objects in front which are from left to right: bunny, tortoise, and chicken. The bunny is 1 unit from the tortoise and the tortoise is 1 unit from the chicken. Imagine the bunny, tortoise and chicken standing on an invisible line. A fourth object yetBaby is 1.0 unit behind the tortoise.

Explain everything that happens in the world when the Run button is pressed.
PART B: Assume the objects are in the same initial starting configuration described in Part A. Note for this part there is no add SceneActivationListener calling myFirstMethod. Consider the following code:

A) Explain everything that happens in for this code when the Run button is pressed and no other keys are pressed.

B) After the program has run for 30 seconds, nothing should be moving. Explain how the user can interact with the program, and what happens when the user interacts with it.
2. (12 pts) Consider the following array of type Flyer called `birds`.

A) For the following code segment, explain what happens when this code executes. Be sure to make it clear who executes an instruction, what happens when they execute an instruction, and the order they execute instructions.
B) For the following code segment, explain what happens when this code executes with the array shown on the previous page. Be sure to make it clear who executes an instruction, what happens when they execute an instruction, and the order the instructions are executed.

C) For the following code segment, explain what happens when this code executes with the array `birds` shown on the previous page plus the additional array `heights` shown below. Be sure to make it clear who executes an instruction, what happens when they execute an instruction, and the order the instructions are executed.
3. (14 pts) **PART A:** Consider the following Scene procedure called `mysteryProc`. Assume `penguins` is an array that has only penguins in it.

![Diagram of the `mysteryProc` procedure]

A) What is `num`, a property or a variable?

B) If there are 8 penguins in the `penguins` array, how many times will the count loop be executed when `mystery` is called?

C) Suppose run is pressed. What must be true for the inner if statement (if `randomnumber == 3`) to never be executed?

D) Explain what the procedure `mysteryProc` does.
PART B: Consider the following Scene function named mysteryFunction that refers to the same penguin array.

A. What type of value does the function mysteryFunction return?

B. Explain what mysteryFunction does.

C. Consider adding the return statement that is marked with the arrow.

Explain what the modified mysteryFunction does.
(extra page 1)
4. (8 pts) Consider the following Alice world that has a **fox** two units from the penguin, and the other animals all within 1 unit of each other: a **tortoise**, a **bunny** and a **penguin**. A mouseClicked event is also shown.

A) When this Alice world runs, what happens when the tortoise is clicked on?

B) When this Alice world runs, what happens when the bunny is clicked on?

C) When this Alice world runs, what happens when the penguin is clicked on?

D) When this Alice world runs, what happens when the fox is clicked on?
5. (10 pts) Consider the following Alice world that has penguins and a tortoise in it. A score of 0 is currently invisible in the top left corner. Shown are the initial picture, some of the events, myFirstMethod and initializing and updating the score. Also shown is the number property (a TextModel property) and the array named penguins (a Scene property).

Answer questions about this program on the next page.
A) Explain what happens when the run button is clicked and nothing else is clicked on.

B) Explain what happens when the run button is clicked and then the user clicks on the tortoise first and then two different penguins.

C) Explain what happens after B) occurs, when the user clicks on the tortoise again.

D) Suppose you add a gameState variable to make the game work better. Explain where and how the gameState variable would be used (which events and procedures and what code would be added).
6. (10 pts) Consider the following Alice world with a cow, camel, horse and wolf, and the addCollisionStartListener. Additional code is not shown.

Answer questions about this code/world on the next page.
A. When the Alice program runs, what happens when the cow runs into the camel?

B. When the Alice program runs, what happens when the horse runs into the cow?

C. When the Alice program runs, what happens when the wolf runs into the camel?

D. Write a new addCollisionStartListener for whenever the cow collides with either of the other three animals, then the cow and the animal it collided with both turn red.
7. (4 pts) Consider the following Alice program where all the animals shown except the panda are in the array named **animals and are in the array in order by height, from shortest to tallest.** The function **firstLargest** is suppose to return the first animal from the array that is taller than the parameter **newAnimal**, that is an animal that is not in the array. If there is no animal taller, then it returns the last animal in the array. When **myFirstMethod** calls the firstLargest function with panda as the argument, the hare says it is the first animal taller than panda, but actually the cheshireCat is the first animal taller!

Explain how to fix the firstLargest function so that it always returns the first animal in the array that is taller than newAnimal, and show the code (you can modify the code above). Your function should work even if the elements in the array are changed but still in order from shortest to tallest.
8. (16 pts) Consider an Alice project with a pig, hare and an array of pandas named pandas as shown on the left below.

A. Write the Scene procedure named `resizeEveryone` that has two parameters. The first one is a `DecimalNumber` named `minValue`, and the second one is a `DecimalNumber` named `maxValue`. This procedure should randomly resize all the pandas in the pandas array some resize value in the range from `minValue` to `maxValue` inclusive. For example, if the call below is made, then the result might be the picture on the right above.

```
this resizeEveryone minValue: 0.25, maxValue: 3.0
```

Complete this procedure below.

```
declare procedure resizeEveryone with parameters DecimalNumber minValue, DecimalNumber maxValue
```
B. Write the Scene procedure called ChangeColor that has been started below for you. This procedure has two Paint parameters named color1 and color2. This procedure randomly changes the color of each panda in the pandas array to either color1, color2, or leaves the panda its current color. For example, for the scene on the left above, suppose the call below to changeColor is made, resulting in the scene on the right above, with each panda randomly colored, red, yellow or not colored.

```
this | changeColor | color1: RED, color2: YELLOW
```

Complete this procedure below.

```
declare procedure changeColor with parameters: Paint color1, Paint color2
```
C. Write the scene Function named `numberOfColorAndShorter` that returns a whole number, and that has been started below for you. This function has two parameters, an SJointedModel named `creature`, and a Paint named `somecolor`. This function returns the number of pandas from the pandas array that are shorter than the creature and are the same color as `somecolor`. For example, with the picture on the right above, if the following three calls are made, there is 1 red panda shorter than the pig, there are two yellow pandas shorter than the pig and there are three yellow pandas shorter than the hare.

Complete the function below.
(extra page 2, must turn in)
(extra page 3, must turn in)