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</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

- `sceneActivated`
  ```java
  declare procedure sceneActivated
do in order
  this myFirstMethod
  ```

- `timeElapsed`
  ```java
  declare procedure timeElapsed = event getTimeSinceLastFire
do in order
  drop statement here
  ```

- `keyPressed`
  ```java
  declare procedure keyPressed
    do in order
    if event isKey is true then
      drop statement here
    else
      drop statement here
  ```

- `mouseClicked`
  ```java
  declare procedure mouseClicked
    do in order
    if event getModelAtMouseLocation == (this.panda) is true then
      drop statement here
    else
      drop statement here
  ```

- `collisionStarted`
  ```java
  declare procedure collisionStarted
    do in order
    drop statement here
  ```
1. (10 pts) **PART A:** Consider the following snapshot of an Alice project in which there are four objects standing on an invisible line from left to right: **panda, pig, hare,** and **bunny.** There is one unit distance between any two consecutive objects. The bunny is facing the hare and the panda is facing the pig.

Explain everything that happens in the world and the order they happen 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 for this code when the Run button is pressed and no other keys are pressed.

B) After the program has run for 20 seconds, nothing should be moving. Explain how the user can interact with the program, and what happens when the user interacts with it.
2. (14 pts) Consider the following array of type Biped named `group` and array of type `Paint` named `someColors`.

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 arrays 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 arrays **group** and **someColors** 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.
3. (16 pts) **PART A:** Consider the following Scene procedure called `mysteryProc`. Assume `pandas` is an array that has only pandas in it.

![Procedure Diagram]

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

B) What is the value of `index` when the loop ends?

C) If there are 10 pandas in the `pandas` array, how many times will the count loop be executed when `mystery` is called?

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

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

B. Explain what mysteryFunction does.

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

Explain what the modified mysteryFunction does.
4. (8 pts) Consider the following Alice project that has four animals shown from left to right they are: cow, wolf, poodle and horse. A mouseClicked event is also shown.

A) When this Alice project runs, what happens when the cow is clicked on?

B) When this Alice project runs, what happens when the wolf is clicked on?

C) When this Alice project runs, what happens when the poodle is clicked on?

D) When this Alice project runs, what happens when the horse is clicked on?
5. (12 pts) Consider the following Alice project with a **flamingo, bunny, panda** and **penguin** (shown left to right), and two events: addCollisionStartListener and addDefaultModelManipulation. Additional code is not shown.

Answer questions about this code on the next page.
A. What does the event `addDefaultModelManipulation` do?

B. When the Alice program runs, what happens when the bunny runs into the penguin?

C. When the Alice program runs, what happens when the flamingo runs into the panda?

D. When the Alice program runs, what happens when the bunny runs into the flamingo?

E. Write a new `addCollisionStartListener` for whenever the flamingo collides with either of the other three animals, then the animal the flamingo collides with fades away and becomes invisible.
6. (12 pts) Consider the following Alice project that has an array of tortoises named tortoises and an array of yetiBabies named yetiBabies. The picture shows a snapshot of playing the game with a timer being displayed as 14 and a scorer being displayed as 3. Also shown are myFirstMethod, initializeTimer procedure, updateTimer procedure, three events, and the TextModel properties, with timerNumber for the timer and scoreNumber for the scorer. The rest of the code is not shown.

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

B) Explain how to modify `updateTimer` with a parameter so the timer can be updated by different amounts.

C) What is the new code for the `timeElapsed` event now that `updateTimer` has a parameter?

D) Give code for a new event so that if you click on any yetiBaby, then 5 seconds is added to the timer. For example, if the timer was currently 15 and the user clicked on a yetiBaby, then the timer would now be 20.

E) The timer continues counting down to -1, -2, -3, etc after it reaches 0. Explain the code needed and where it goes to make the timer stop counting down once it reaches 0.
7. (6 pts) Suppose a student has written a game that has a wolf and an array of type bunny named bunnies. Whenever the wolf collides with a bunny from the bunnies array, that bunny is supposed to disappear by moving quickly below ground and the score is supposed to be updated. Three points are supposed to be added to the score if that bunny was painted red, and otherwise one point is supposed to be added to the score. Assume the procedure increaseScore works correctly, adding the amount passed in to the score.

Consider the code below that does not work correctly.

A) Explain what happens when this code runs and the wolf collides with a bunny.

B) Explain how to fix the code so it works correctly.
8. (18 pts) Consider an Alice project with an array of type bunny named bunnies that has bunnies of all sizes as shown on the left below.

A. Write the Scene procedure named \texttt{resizeSmallOnes} that has three parameters. The parameters are three DecimalNumbers named \texttt{heightCutoff}, named \texttt{minNumber}, and named \texttt{maxNumber}. This procedure should randomly resize those bunnies in the bunnies array whose height is less than \texttt{heightCutoff}. They should be resized by a random value in the range from \texttt{minNumber} to \texttt{maxNumber} inclusive. For example, if the call below is made, then the result might be the picture on the right above.

```
this resizeSmallOnes heightCutoff: 0.75 , minNumber: 1.5 , maxNumber: 2.5
```

Complete this procedure below.
B. Write the Scene procedure named `changeColor` that has been started below for you. This procedure has two parameters. One is of type `wholeNumber` and is named `amount`, and the other is of type `Paint` and is named `someColor`. This procedure paints the first `amount` number of bunnies in the `bunnies` array the color `someColor`. 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 the first three bunnies in the `bunnies` array colored red.

Complete this procedure below.

```
C. Write the scene Function named `howManyTheseColors` that returns a whole number, and that has been started below for you. This function has two Paint parameters named `color1` and `color2`. This function returns the number of bunnies from the bunnies array that have been painted the color `color1` or `color2`. For example, in the picture above there are three red bunnies, five yellow bunnies and two green bunnies. In the first call below the bunny says number red and yellow are 8, and in the second call the bunny says number green and yellow are 7.

Complete the function below.