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.
1. (3 pts) Which one of the following HTML tags does not result in bold letters?

   A) `<b>`  
   B) `<em>`  
   C) `<h2>`  
   D) `<strong>`

2. (3 pts) Consider the following list that has been created with HTML.

   ○ UNC  
   ○ NCSU  
   1. WFU  
   2. Duke

Which one of the following is the HTML that generated this list?

   A) `<ul>`  
      `<li>` UNC `</li>`  
      `<li>` NCSU `</li>`  
      `<ol>`  
      `<li>` WFU `</li>`  
      `<li>` Duke `</li>`  
      `</ul>`

   B) `<ol>`  
      `<li>` UNC `</li>`  
      `<li>` NCSU `</li>`  
      `<ul>`  
      `<li>` WFU `</li>`  
      `<li>` Duke `</li>`  
      `</ul>`

   C) `<ul>`  
      `<ol>`  
      `<li>` UNC `</li>`  
      `<li>` NCSU `</li>`  
      `</ol>`  
      `<li>` WFU `</li>`  
      `<li>` Duke `</li>`  
      `</ul>`

   D) `<ol>`  
      `<li>` UNC `</li>`  
      `<li>` NCSU `</li>`  
      `</ol>`  
      `<li>` WFU `</li>`  
      `<li>` Duke `</li>`  
      `</ol>`
3. (3 pts) Consider the following HTML code. Draw what it would look like on a web page.

```html
<table border=1>
  <tr>
    <td> A </td><td> B </td>
  </tr>
  <tr>
    <td> C </td><td> D </td>
  </tr>
  <tr>
    <td> E </td><td> F </td>
  </tr>
</table>
```

4. (3 pts) Consider the following CSS for table.

```css
table {
  color: red;
}

table td {
  color: blue;
}
```

Explain how these two will affect how a table is displayed.
5. (4 pts) Consider the following that appears in a .css file.

```css
#largeFont {
    font-size: 64px;
}
```

a) Is this an ID or a Class?

b) Consider the .html file that refers to this .css file. Give the .html that shows how one applies the CSS above to the word “football” in the following line in the .html file.

```
Where are the football fans?
```

6. (4 pts) A six-digit hexadecimal number represents a color. A color can also be represented as a three-tuple of numbers representing the red, green and blue components (or RGB) of the number. You must show your work for full credit!

   A. Convert the hexadecimal number 061B23 into RGB: (            ,         ,         )

   B. Convert the color in RGB (50, 13, 27) to the corresponding six-digit hexadecimal number:
7. (3 pts) Consider the following Alice code and the starting position of the whiteRabbit and panda that is shown with Start in the figure on the left below.

Which one of the following shows the result after starting in the start position (on the left) and then executing this Alice code?

![Options A, B, C, D]

8. (3 pts) Consider the following Alice code and the starting position of the whiteRabbit and panda that is shown with Start in the figure on the left below.

Which one of the following shows what the whiteRabbit looks like starting in the start position (on the left) and then executing this Alice code?

![Options A, B, C, D]
9. (3 pts) Consider the following Alice code and the starting position of the whiteRabbit that is shown with **Start** in the figure on the left below.

Which one of the following shows what the whiteRabbit looks like starting in the start position (on the left) and then executing this Alice code?

```
do in order
  do in order
    this.whiteRabbit turn RIGHT, 0.25
    add detail
  this.whiteRabbit roll RIGHT, 0.25
  add detail
```

A) ![Image A]
B) ![Image B]
C) ![Image C]
D) ![Image D]
10. (14 pts) Consider the following Alice code in which the lines are numbered.

A) In line 1, is “something” a function or a procedure?

B) In line 2, what TYPE is “this.bunny isBehind this.pig”?

C) In line 3, what is the name of the function and what TYPE does it return?

D) What must be true so that line 5 executes?

E) In line 6, list the word(s) that are arguments.

F) In line 6, list the word(s) that are parameters.

G) In line 6, is “tricky” a function or a procedure?
11. (6 pts) Consider the following world that has the three objects: stuffedTiger, marchHare and panda (shown below from left to right) and the given code. The world has been setup as shown below. The stuffedTiger is exactly 1.0 meter from the marchHare, and the panda is exactly 1.0 meter from the marchHare.

The diagram below is looking from above over the scene. The stuffedTiger is represented by the S, the marchHare is represented by the M, and the panda is represented by the P. The animals are facing the bottom of the page. Using the diagram below, draw the path of stuffedTiger and marchHare as a solid line and the path of panda as a dashed line.
12. (4 pts) Consider the following Mystery function.

A) What does pig say when the following line of code is run?

B) What does pig say when the following line of code is run?
13. (10 pts) Consider the following Alice world that has three objects: cow, cedarPole and camel.

The world starts as shown in the figure above with the cow and camel facing each other and also facing the cedarPole. Assume there is a cameraMarker called cameraStart that is the position the camera is currently shown in. When you move the animals you do not need to move their legs, just move them. Write code to do the following in this order.

a) Move the camera so it is above looking down on the animals.
b) Have the cow and camel turn twice around the cedar pole at the same time, going the same direction and moving with their head first (not sideways).
c) Move the camera back to the camera starting view.

```
declare procedure myFirstMethod
  do in order
```
14. (9 pts) Complete the following Alice **procedure** called `GetInLine` whose header is shown on the next page. This procedure is called several times to build a line of creatures lined up behind Alice. In particular, this Alice procedure has three parameters, an SJointedModel named “creature,” a decimal number named “howFar,” and an SJointedModel named “lastCreature”. Assume the Alice world starts with Alice facing three creatures who are lined up, all facing the camera as shown below. This procedure has `creature` line up behind Alice, more specifically placed “howFar” behind the “lastCreature” who is the last one in line. Once the creature is in place, code is set so that if Alice moves, the creature will also move.

Here are three calls to this procedure followed by Alice moving forward with everyone following.

![Image](image1.png)

The result of this code is that the tortoise moves to 0.5 behind Alice, then the panda moves to 0.5 behind the tortoise, who is last in line, then the bunny moves to 1.0 behind the panda, who is last in line. Then in the last line of code Alice moves forward and everyone in line follows her.

![Image](image2.png)
Complete the code below.

```
declare procedure getInLine with parameters: SJointedModel creature,
                                DecimalNumber howFar, SJointedModel lastCreature
```
15. (10 pts) Assume there are four objects in an Alice world, a mapinguari, a bunny, a panda, and a blackCat (shown in that order left to right below) all facing the camera. Complete the following bunny function called fitBetween that has two STurnable parameters, one named friend, and one named object. This function returns True if object can “fit between”, or stand between bunny and friend, while the object is also facing the camera, and returns False otherwise.

Hint: Note that the “distanceTo” function measures the distance between two objects from the center of their bodies, the bottom double arrow below. You want to measure the distance between them, the top double arrow below.

Complete the function on the next page.
A) (7 pts) Complete the **bunny function** below.

```plaintext
declare Boolean function fitBetween(sJointedModel object friend, sJointedModel object object)
```
B) (3 pts) Give the code to have the blackCat say “I fit”, if it can fit between the bunny and panda, and to say “I don’t fit” if it cannot fit between them. Use your function to determine this. Then give the code to have the mapinguari say “I fit”, if it can fit between the bunny and panda, and to say “I don’t fit” if it cannot fit between them. Use your function to determine this.