Given below are the world functions.

Tiles at the bottom of the Alice window.

Events:

Not shown: When something is true
Format of instructions:

- If true
  - Do Nothing

- Else
  - Do Nothing

- Loop 5 times
  - Times
  - Show complicated version
  - Do Nothing

- Loop 123 index from 0 up to (but not including) 5 times, incrementing by 1
  - Do Nothing

- While true
  - Do Nothing

- For all world: horses, one item from horses at a time
  - Do Nothing

- For all world: horses, every item from horses together
  - Do Nothing

Chicken properties and methods:

- Chicken's details
  - Properties
  - Methods
  - Functions
  - Create new variable
  - Capture pose
  - Color
  - Opacity = 1 (100%)
  - Vehicle = world
  - Skin texture = Chicken.TextureMap
  - FillingStyle = solid
  - PointOfView = position: 0, 0, -0.1; orientation: (0, 0, 0) 1
  - IsShowing = true

- Chicken's details
  - Properties
  - Methods
  - Functions
  - Create new method
  - Chicken move
  - Chicken turn
  - Chicken roll
  - Chicken resize
  - Chicken say
  - Chicken think
  - Chicken play sound
  - Chicken move to
  - Chicken move toward
  - Chicken move away from
  - Chicken orient to
  - Chicken turn to face
  - Chicken point at
  - Chicken set point of view to
  - Chicken set pose
  - Chicken stand up
  - Chicken move at speed
  - Chicken turn at speed
  - Chicken roll at speed
  - Chicken constrain to face
  - Chicken constrain to point at
Given below are the chicken functions.

Array Visualization special methods and functions
1. (12 pts) Consider the following snapshot of an Alice world in which there are four objects which are (from left to right) duckPrince, magicBunny, and toadstool standing on circle, which has been colored red. The duckPrince and magicBunny are 3 meters apart. The magicBunny and toadstool are 3 meters apart. Imagine the three of them are standing on an invisible line. Note that the ToadStool height is 1.7.

A) List the events in this world.

B) List the event handlers in this world.

C) Explain what happens in the world when the Play button is pressed.

D) How many total times will the duckPrince circle the magic Bunny?
2. (4 pts) Consider the following array of animals shown below as a picture and as a property, and the following code. Also shown in the picture are an ArrayVisualization and an ObjectVisualization.

Here is the code:

Remember that when an object is printed, its name is printed. What output is printed when this code is executed?
3. (10 pts) Consider the following function called `world.mystery`. Assume a list called `world.penguins` has been created and filled with penguins.

   A. What type of value does the function `world.mystery` return?

   B. What is `temp`: a local variable, a class variable or a parameter?

   C. What is `something`, a local variable, a class variable or a parameter?

   D. Explain what the function `world.mystery` does.

   E. If the statement “temp set value to item_from_penguins” is replaced with the statement “return item_from_penguins” does the function do the same thing? Explain.
4. (6 pts) Consider the following Alice code.

A. How many times does the carousel turn one complete turn?

B. How many times does horse2 move up and down (count once for moving up and then down)?

C. How many times does shannon wave (raise and lower her arm)?
5. (10 pts) Consider the following Alice world, which includes a 3DText called timer, and a few basketballs for a game shown in the picture on the left below. There is a BDE event to decrement the timer and the code for decrement is also shown.

Answer questions about this world on the next page.
A. Write the method `timer.cheat` that will increment the timer’s variable `timer.value` by 10 only if its current value is 10. The new score value must be displayed in the world. For example, if `timer.value` is 15 when `timer.cheat` is called, then its value is unchanged. If `timer.value` is 10 when `timer.cheat` is called, then its value changes to 20.

B. Give the event for the following. When the user presses “C” the cheat method should execute.

C. One problem is that the timer can display negative values. What code should be added/modified so that the timer will stop when it reaches 0?
6. (8 pts) Write the method world.meeting that has no parameters.

In this method there are two objects, AliceLiddell and alienOnWheels that are apart and facing each other. Alice Liddell says “Come closer, please”. The alienOnWheels moves forward either -1, 0, 1 or 2 units. This repeats with Alice asking the alienOnWheels to come closer and the alienOnWheels randomly moving an integer unit between -1 and 2 units inclusive until the alienOnWheels is closer than 1.5 units to AliceLiddell. At that point she says “Hello”. Note that you do not want the alienOnWheels to run into Alice, so if its random amount to move will move it closer than 1 meter to Alice, then instead just move it so it stops 1 meter in front of Alice.
7. (8 pts) Write the function `world.numberOutOfOrder` that has no parameters. This function returns the first number in the array `values` that is out of order (that is, it is the first number in the array that decreases in value from the previous element). For example, the picture on the right below shows the array `values`. The call to `numberOutOfOrder` returns 8. Note that the first three numbers are in increasing order: 3, 7, 15. Then 8 comes next, which is a decrease in value from 15.

Write your code below. Your code should work regardless of the numbers and size of values. That is, your code should still work if numbers in `values` are changed, or new numbers are added to the array.
8. (20 pts) Assume all the chickens below are in a list called `all`. Note that the chickens are of different heights.

A. Write the function `world.tallest` with one parameter: `chickcolor`. This method returns the tallest chicken that has the color `chickcolor`. Complete the method below.
B. Write the function `world.PickFirstInList` that has one parameter `somecolor` and returns the first chicken in the list with color `somecolor`. Complete the function below.

```
# world object

d = world.PickFirstInList  # PickFirstInList is a method defined later.

d = world.PickFirstInList(somecolor=123)  # Example usage with color 123.
```

C. Write the method `world.pickTeams` that has three parameters, a color called `teamAColor`, a color called `teamBColor` and a number `size`. This method should alternate choosing chickens from the list `all` and changing their color to their team color, picking “size” chickens for each team. Team A always chooses the tallest chicken of `noColor` when it selects a chicken. Team B is not so choosy, it just selects the first available chicken that is `noColor`. For example, if size is 2, then team A would go first and pick the tallest Chicken with `noColor` and color it the color `teamAColor`. Then `teamB` would pick the first chicken in the list that is `noColor` and color it the color `teamBColor`. Team A would go again and then Team B so they would both have a team of size 2. Assume there are enough chickens of color `noColor` to be picked for each team. **You must call the functions** you wrote in parts A and B to receive full credit.

```
# world object

d = world.pickTeams(teamAColor=123, teamBColor=456, size=2)  # Example usage with teams A and B and size 2.
```
9. (14 pts) Consider a world with the arrayVisualization of animals shown below. You may use AV as an abbreviation for arrayVisualization.

A) Write the function `world.indexOutOfOrder` that returns the index of the first animal in the array whose height is out of order (that is the index of the first person who is shorter than animals earlier in the array). If everyone is in increasing order, then return -1 instead.

B) Write the method `world.fixOrder` that increases the size of any animal in the array that is out of order based on its height. The size is increased by 10%. This is done repeatedly until the animal is taller than all the animals that appear earlier in the array.