1. What does RGBA stand for?

A. RGBA stands for Red, Green, Blue, and Alpha.

2. From the Pillow module, how do you get the RGBA value of any images?

A. In order to get the RGBA value of any image using the Pillow module in Python, we need to open the image using the **Image** module, and then use the **getpixel()** method to get the pixel values at a specific coordinate.

This will output a tuple with four integer values, representing the red, green, blue, and alpha values of the pixel, in that order.

3. What is a box tuple, and how does it work?

A. In Pillow module, a box tuple is a tuple of four integers representing the left, upper, right, and lower pixel coordinate boundaries of a rectangular region. The format of a box tuple is (left, upper, right, lower), where left is the x-coordinate of the leftmost pixel, upper is the y-coordinate of the topmost pixel, right is the x-coordinate of the rightmost pixel (not included), and lower is the y-coordinate of the bottommost pixel (not included).

Box tuples are commonly used to represent rectangular regions of an image or to crop images. For example, to crop an image to the region between the coordinates (100, 100) and (200, 200), we can create a box tuple **(100, 100, 200, 200)** and pass it to the **crop()** method of the **Image** object.

4. Use your image and load in notebook then, How can you find out the width and height of an Image object?

A. To find out the width and height of an Image object, we can use the **size** attribute, which returns a tuple containing the width and height of the image. Example:

img = Image.open('example.jpg')

width, height = img.size

print(f"Width: {width} pixels")

print(f"Height: {height} pixels")

5. What method would you call to get Image object for a 100×100 image, excluding the lower-left quarter of it?

A.   
To get an Image object for a 100x100 image excluding the lower-left quarter of it, we would first open the image using the **Image.open()** method and then crop the image using the **Image.crop()** method with a box tuple of **(0, 50, 50, 100)**, which specifies the region from (0,50) to (50,100) pixels.

6. After making changes to an Image object, how could you save it as an image file?

A. To save changes made to an Image object as an image file, we can use the **save()** method of the Image object. The **save()** method takes the filename of the image file to be saved, as well as an optional format parameter.

7. What module contains Pillow’s shape-drawing code?

A. The **ImageDraw** module contains Pillow's shape-drawing code.

8. Image objects do not have drawing methods. What kind of object does? How do you get this kind of object?

A. The **ImageDraw** module contains Pillow's shape-drawing code. We can create an **ImageDraw** object for a given Image object by calling the **ImageDraw.Draw()** function with the Image object as an argument. The **ImageDraw** object has drawing methods that can be used to draw on the Image object. Example:

image = Image.open('example.jpg')

draw = ImageDraw.Draw(image)

draw.rectangle((10, 10, 50, 50), fill='red')