Solutions:

1. This is mostly about understanding indexing, slicing, and the order of operations of expressions
   (a) `1729` - Recalling order of operations
   (b) `False` - Precendence order shown in Figure 1-1 in text
   (c) `FILO`
   (d) `5` - You don't really need to recall what `ord("F")` is here, just that all the capital letters come in a block, so whatever `ord("F")` is, it has to come 5 values after `ord("A")`.

2. This is all about following along with a program, and understanding the scoping of variables. You can not use PythonTutor on the test, so write out what the state of the variables looks like at each point!

```
b
bagg
baggage
```

3. As is usual with this sort of thing, there are a variety of ways this could have been approached. This way is just one method.

```
def longest_character_run(s):
    longest_sub = ""
    current_sub = s[0]
    for i in range(1, len(s)):
        if s[i] == current_sub[-1]:
            current_sub += s[i]
        else:
            current_sub = s[i]
        # Check bigger, so ties remain with first occurance
        if len(current_sub) > len(longest_sub):
            longest_sub = current_sub
    return longest_sub
```

4. Same applies to this code, but this is how I'd approach it. I didn't waste time showing any import statements, as they can be assumed. The way the 3rd stipulation was worded, I wasn't clear if that random direction should change every 20 milliseconds or not, but then 4 makes me think that clicking is the only thing that should change it. So that is how I wrote the below.

```
def make_cross():
    cross = GCompound()
    hrec = GRect(60, 20)
    hrec.set_filled(True)
    hrec.set_color("Red")
    cross.add(hrec, -hrec.get_width()/2, -hrec.get_height()/2)
    vrec = GRect(20, 60)
    vrec.set_filled(True)
    vrec.set_color("Red")
    cross.add(vrec, -vrec.get_width()/2, -vrec.get_height()/2)
    return cross

def move():
    cross.move_polar(2, gs.angle)

def click(e):
```

```
17      x = e.get_x()
18      y = e.get_y()
19      if gw.get_element_at(x,y) is not None:
20          gs.angle = random.randint(0,360)
21
22  gw = GWindow(200,200) # Size doesn't matter here apparently
23  gs = GState()
24  gs.angle = random.randint(0,360)
25  cross = make_cross()
26  gw.add(cross, gw.get_width()/2, gw.get_height()/2)
27  gw.set_interval(move, 20)
28  gw.add_event_listener("click", click)
```