**ANSWER KEY**

# CODEBOT MISSION 4 LOG - Lesson 2

| **Pre-Mission Warm-Up** | |
|---|---|
| What line of code increments a counter? | Answer should be code, and any variable can be used:<br>`count = count + 1` |
| How can you avoid a ValueError? | Answers can be either explained, or shown with code:<br>To avoid a ValueError, the number of an LED must be within the correct range. For user LEDs, the value must be between 0 and 7.<br>`if n_led == 8:`<br>`  n_led = 0` |
| **Mission 4 Lesson 2 – Animatronics** | |
| **Mission 4 Introduction (Review)** | |
| What is the second step outlined in the notes? | Each guest presses a button as they enter the small entrance room. When 5 guests have entered, the show starts! |
| What is the second project goal? | Blink the red user LEDs constantly in a cool pattern. |
| **Mission 4 Objective 4**<br>• Create a new file ***CountGuests***. Follow the slides to write your code. | |
| How can you stop an infinite loop? | By using a break statement. |
| **Mission 4 Objective 5** | |
| What does it mean to increment a variable? | Add one to it. |
| What is the code for turning on a line sensor LED, using a variable? | `leds.ls_num(n_guests, True)` |
| After you add code to the program and run it, what did the program do? | The CodeBot starts off without any LEDs on. Then, each time BTN-0 is pressed, a green line sensor LED turns on. When all five are on, the program stops.<br><br>Students may observe that one button press actually lights up two LEDs. This is a button bounce, and will be addressed in Obj. 7. If students bring it up now, refer back to their comments in Obj. 7. |
| **Mission 4 Objective 6** | |
| What functions turn on and off the CodeBot speaker? | `spkr.pitch()` - with a frequency in Hertz<br>`spkr.off()` |
| **Mission 4 Objective 7** | |

| | |
|---|---|
| How do you keep the buttons from bouncing? | Add in a short delay, and then reset the internal status by calling the buttons.was_pressed() function again. |
| **Mission 4 Quiz** | |
| Take the quiz. Which questions did you miss? Are there any concepts you need to review? | Answers will vary.<br><br>The third question has a while loop that isn't infinite. It has the condition i < 3. This hasn't been covered yet, and the variable i is incremented outside the loop, so it becomes an infinite loop. You may need to go over this question. |
| **Post-Mission Reflection** | |
| Explain the steps for counting guests. | Answers can vary. The answer may be a couple of sentences that discuss this, or a step-by-step algorithm.<br><br>To count guests, you need a variable to keep track of the button presses. Increment it each time the button is pressed. Then break the loop when the number of guests is reached.<br>• Define a count variable (n_guests) and initialize to 0<br>• Use an infinite loop<br>• Use an if statement to check for a button press.<br>• If the button is pressed<br>• Turn on the next line sensor LED<br>• Increment the count variable (n_guests)<br>• Compare the count variable to the max number.<br>• If it is the max number, break the loop |
| What does it mean to "debounce" a button? | Answers can vary. Possible answer:<br><br>To debounce a button, you need to wait for the metal contact to settle down before checking again. Use these steps:<br>• Detect a bounce<br>• Wait for the contact to settle<br>• Reset the internal status to False by calling buttons.was_pressed() again |