day97 micro:bit falling dots
Thursday 5/12/22 (6th period), Friday 5/13/22 (7th period)

Download the file "falling_dot_starter_code.py" from the Google Classroom, load it into the Mu Editor, and save it with your name in the comment at the top. Upload it to your micro:bit.

The code lights up a single pixel in the center column (column 2) briefly, then turns it off, then lights up the one below it, and repeats, making it look like a falling dot or drop of water.

Step 1: rewrite this code so that you use a for loop instead of 5 copies of the light on/off code. See if you can figure out where you would put the for line, and what it would say to make a variable that would go from 0 to 4 so that you didn't need five copies of the block that turns a light on, sleeps, then turns the light off.

Step 2: Change the code so that the dot slowly goes faster and faster over time, never going faster than a sleep of 50 milliseconds.

Step 3: Change the code so that if you press button A the falling dot moves one column to the left and goes back to the starting speed (again, it automatically speeds up as it progresses). Add code to ensure that the micro:bit does not crash if you try to move past column 0. Use "if button_a.was_pressed():".

Step 4: Change the code so that if you press button B the falling dot moves one column to the right. Make the same checks to make sure it does not crash if you press the B button and it is already on the far right side.

Step 5: Add code to count how many times the bottom, center LED gets turned on and off (that is column 2, row 4). In between dropping dots make it light up to the brightness of its count, meaning, if it hasn't been ever reached it stays off, if it has been reached once it stays lit at power level 1 instead of being turned off after the main dot has dropped down to it and then started back up at the top, and so on. In other words, if you reset the micro:bit, the dot will drop 9 times and that LED will slowly go from zero to power 9, steadily.

Step 6: Add code that implements the bottom dot count for the other four columns. I did it using a list (I started the list equal to [0,0,0,0,0], one zero for each bottom LED count.) You can do it however you want.

Step 7: Add code to stop the falling drop and put up an image or a short animation if you got all the bottom dots to be power 9, stay there forever.

Call me over to check you off when you are done. I strongly recommend that you do the steps in order. You could even save your step 1 as Step_1.py, then select all, and paste into a new document, do step 2, and save as "Step_2.py", etc. so that you can always go back and look at your last working stage if you run into trouble. This is a complicated project!

For extra credit: add something else totally cool that enhances the experience. Do the dots change direction (they could go left to right, for example, after the bottom has all been lit up…) Do you "fill up" the whole LED array drop by drop like a glass of water? Have some fun with it.