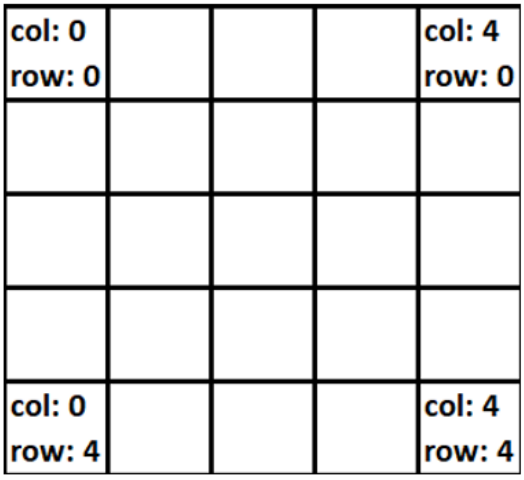
day76 micro:bit pixel art
Due Monday 3/13/23

The lights in your image are listed in the lights list with the column first, then the row



Write a program that draws an image on the 5 by 5 grid, one pixel at a time. You will encode the image in a list variable, and the user will "draw" the image, one light at a time, by pressing the A button. Each time they press the A button another light of your image will light up.

First, make a count variable set to zero.

Then make a list variable that contains LED locations like this:

```
lights = [ ( 2, 2), ( 2, 3), ... ]
```

Each ordered pair represents a light that you will turn on to draw a custom image. Each pair of numbers is a column, and then a row. So for example, the top right light is column 4, row 0. The column is the first number.

Create an empty image, today let's call it "pic":

```
pic = Image()
```

Add code to light up the lights using your count variable like this:

```
col = lights[count][0]
row = lights[count][1]
pic.set_pixel(col,row,9)
display.show(pic)
```

Add code to add one to the count variable each time you press the A button, and you'll be "drawing" an image before you know it.

Add code to reset the image if you press the B button (reset the image, plus set count back to 0). All the lights should be off.

Your image must feature at least 15 pixels and it must be something cool, interesting, fun, and recognizable.

Add code to check if count has reached its maximum and do not add one anymore at that point (otherwise the micro:bit will crash when you press it one too many times.)

Do the above to get 7/10 points.

Continue on the next page for the rest.

To get full credit, add a power code to each ordered pair that says how bright to make each light and then adapt your code to use it, setting some of the lights to not be at full power. For example, instead of using just (2, 2) for a light you could use (2, 2, 5) to indicate that you want to set the light at 2, 2 to intensity 5. You can then access the intensity like this: p = lights[count][2] and then use p in the set_pixel call instead of 9.

Test your code before you call me over to see your work. If you keep clicking the A button after your drawing is done, does it crash? Does pressing the B button clear the image? Does your image have 15 or more pixels lit? Is your image cool, fun, recognizable and interesting? If you did the second part, are you using different power levels for your lights?

Call me over to see your work once you are done. Turn in your code on the Google Classroom.