**Code conversation full solution - whisp-o-meter**

**Code for program is here (called shoutometer)**

**https://github.com/monkmakes/micro\_bit\_kit**

**Describe what the whole program is doing in a couple of sentences:**

If you make a noise near the microphone section of the MonkMakes Sensor Board the LEDs on the micro:bit light to show the volume. The louder the noise, the more LEDs will be lit. The program defines a function called bargraph which firstly clears the LED display then produces a bar graph representing the volume of received sound waves on the LED display.

**Fill in the blanks**

|  |  |
| --- | --- |
| ***Code*** | ***What is this line doing?*** |
| from microbit import \* |  |
| This function will determine the bargraph display on the micro:bit |  |
| def bargraph(a): |  |
| display.clear() |  |
| for y in range(0, 5): |  |
| if a > y: |  |
| for x in range(0, 5): |  |
| display.set\_pixel(x, 4-y, 9)\*\* |  |
| while True: |  |
| sound\_level = (pin0.read\_analog() - 511) / 100\* |  |
| bargraph(sound\_level) |  |

\*Since pin0 is connected to the microphone, the signal will vary between 0 and 1023 as the sound wave oscillates. 511 is subtracted from the reading to get a midpoint of the signal. This means that sometimes the result to be displayed by the plot bar graph blocks will be negative, but that doesn’t matter as the bar graph plotter will just ignore those values.

\*\*display.set\_pixel(x,y,value) - this inbuilt function sets the brightness of the LED at row x and column y to value, which has to be an integer between 0 and 9. In this case the brightness is always set to the maximum 9.