Girl Power in Engineering and IT

Introduction to Microbits

Wednesday 5 July 2023

Created by Rama Kaorma

POSTERA CRESCAM LAUDE

THE UNIVERSITY OF MELBOURNE

SOLUTIONS

Part 0: Getting Started

Task 0.1: Give your program a name

At the top of your browser, you should see an ‘Untitled project’ box. Give your project a name.

Task 0.2: Write your name

In your microbit editor, add a comment at the top with your name. This is to say that you have written this program!

Task 0.3: Import your library

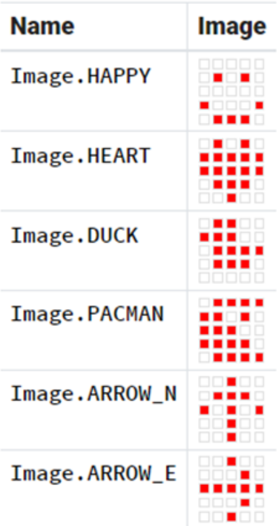
In your microbit editor, add a statement to import the library “microbit”. We want to import **all** the contents of that library. Note that this might already be done for you

Try sending the current code to your microbit, what do you notice?

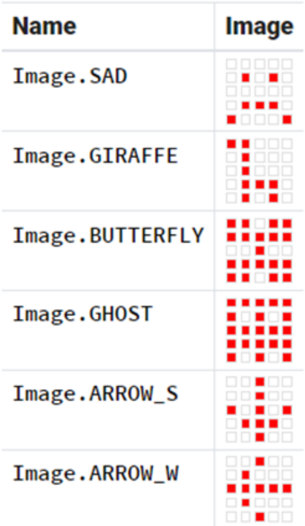
Part 1: Start with a smile :)

Task 1.1: Display your first image

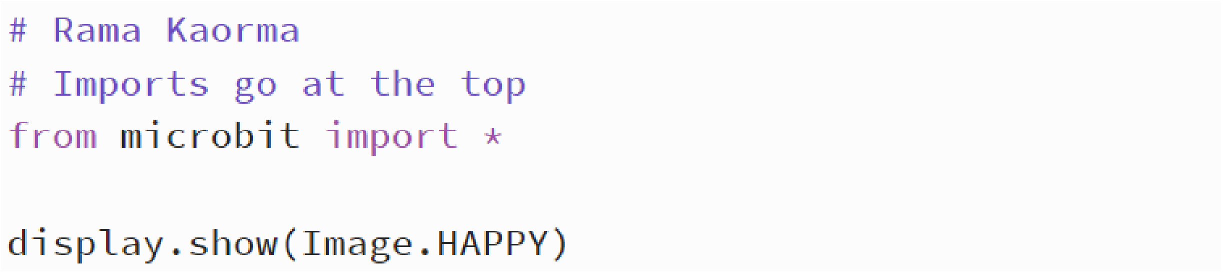
Display an image of your choice based on the list below:



Name	Image
Image.HAPPY	
Image.HEART	
Image.DUCK	
Image.PACMAN	
Image.ARROW_N	
Image.ARROW_E	



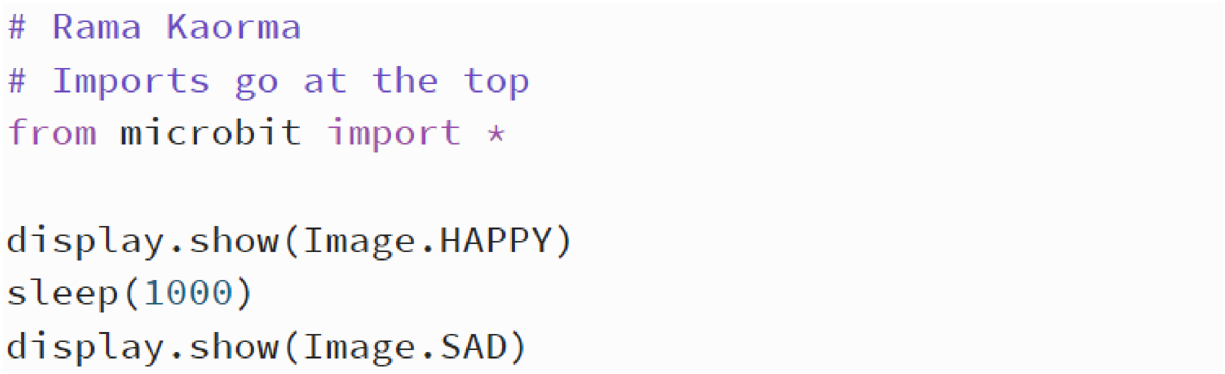
Name	Image
Image.SAD	
Image.GIRAFFE	
Image.BUTTERFLY	
Image.GHOST	
Image.ARROW_S	
Image.ARROW_W	



```
# Rama Kaorma
# Imports go at the top
from microbit import *

display.show(Image.HAPPY)
```

Task 1.2: Two in a row

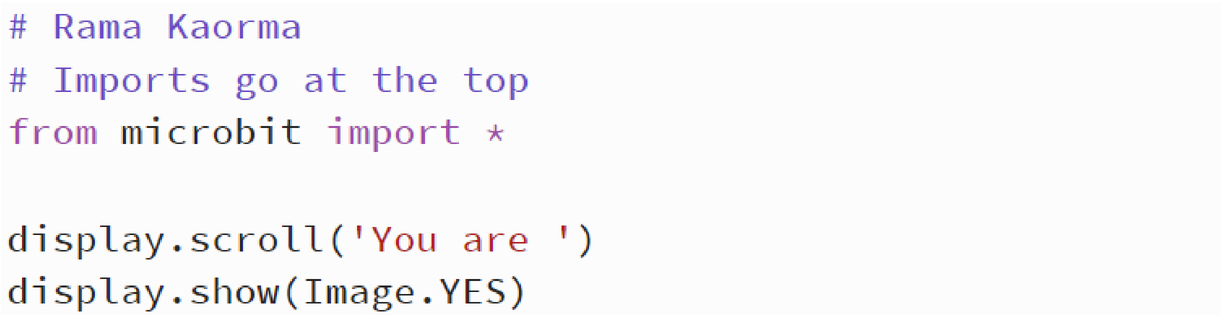
Display multiple images of your choice (in a row) based on the previous list. Remember to pause.



```
# Rama Kaorma
# Imports go at the top
from microbit import *

display.show(Image.HAPPY)
sleep(1000)
display.show(Image.SAD)
```

Task 1.3: You are correct!

Display the message “You are ✓”. The ✓ is displayed using “Image.YES”

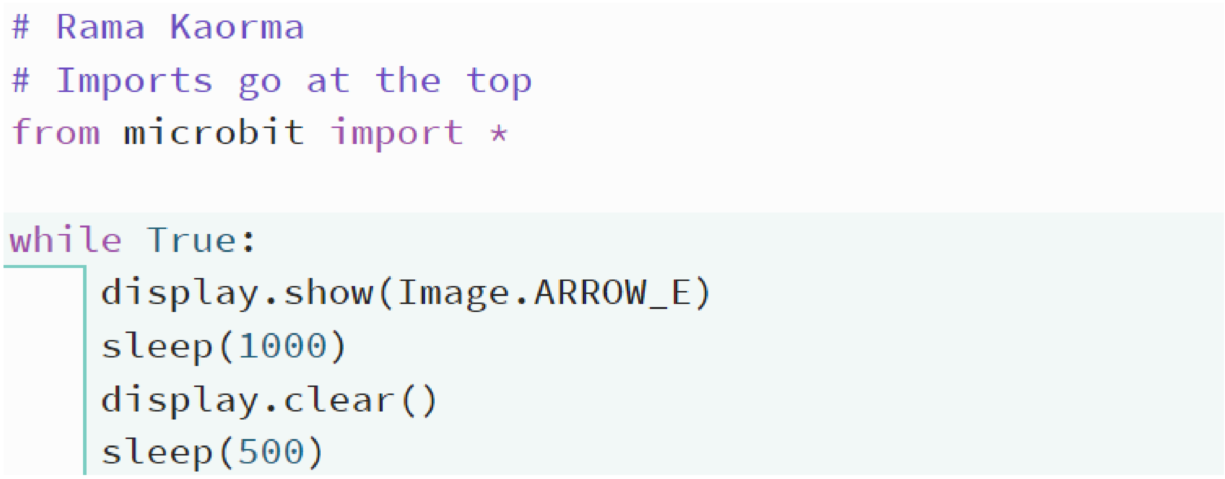


```
# Rama Kaorma
# Imports go at the top
from microbit import *

display.scroll('You are ')
display.show(Image.YES)
```

Part 2: Don’t stop

Task 2.1: Merge lane

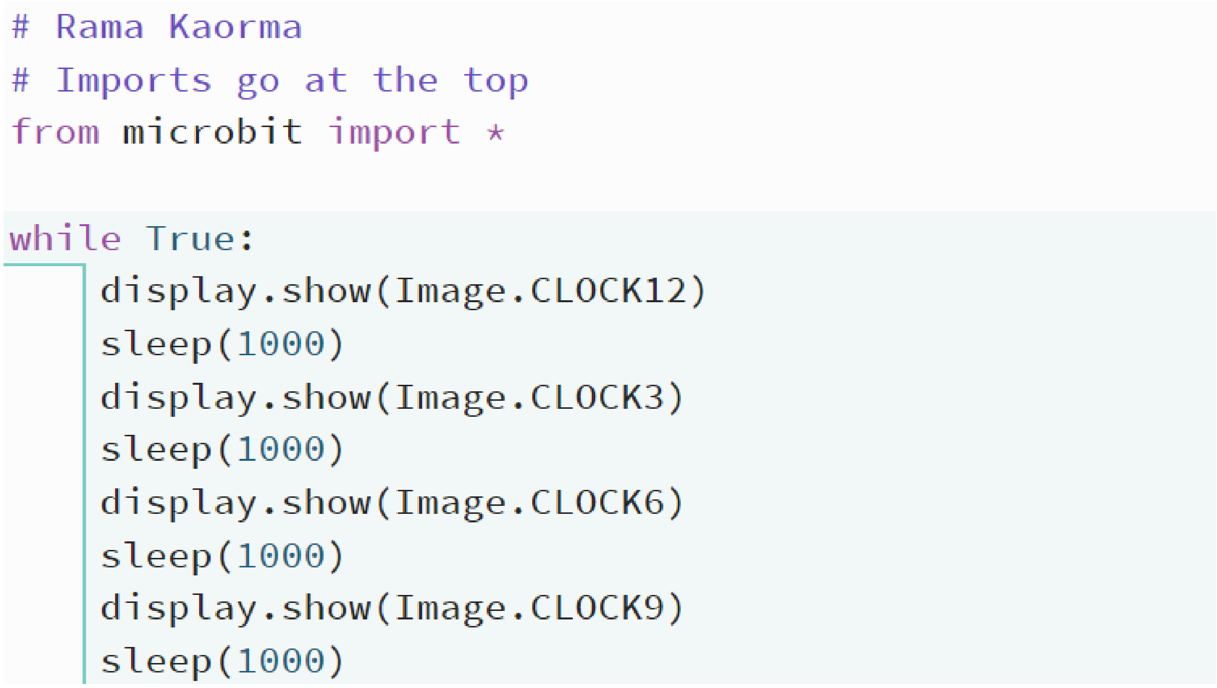
Write a program to redirect traffic. Your program should flash Image.ARROW_E continuously, staying on for 1 second, and off for half a second.



```
# Rama Kaorma
# Imports go at the top
from microbit import *

while True:
    display.show(Image.ARROW_E)
    sleep(1000)
    display.clear()
    sleep(500)
```

Task 2.2: Tick Tock.. Not Tiktok

Time is ticking! Write a program to move a clock hand continuously around the display. It should start at 12 o'clock, go to 3 o'clock, then 6 o'clock, and then 9 o'clock, staying in each position for one second. Use the following as a guide.

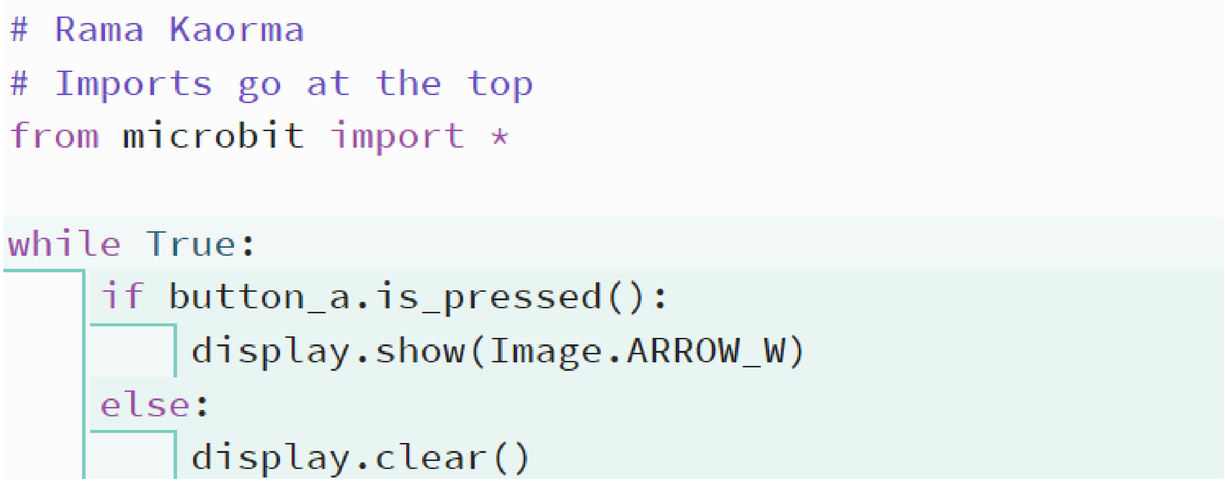


```
# Rama Kaorma
# Imports go at the top
from microbit import *

while True:
    display.show(Image.CLOCK12)
    sleep(1000)
    display.show(Image.CLOCK3)
    sleep(1000)
    display.show(Image.CLOCK6)
    sleep(1000)
    display.show(Image.CLOCK9)
    sleep(1000)
```

Part 3: Buttons and conditions

Task 3.1: Point at

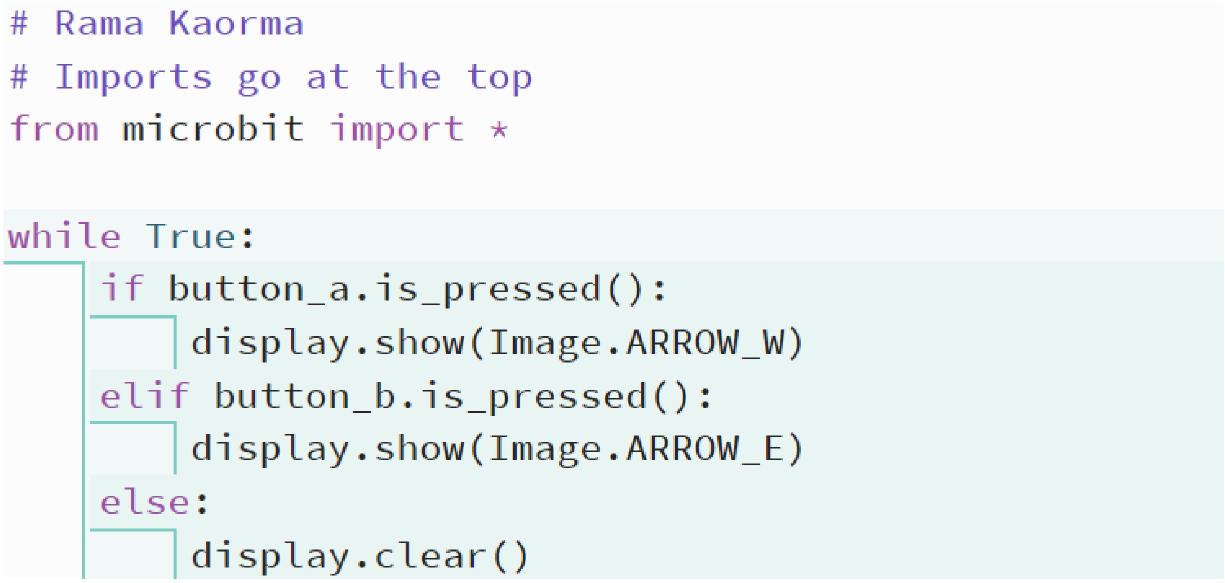
Write a program that points to button A when button A is pressed. Otherwise, clear the display.



```
# Rama Kaorma
# Imports go at the top
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.ARROW_W)
    else:
        display.clear()
```

Task 3.2: Point at - continued

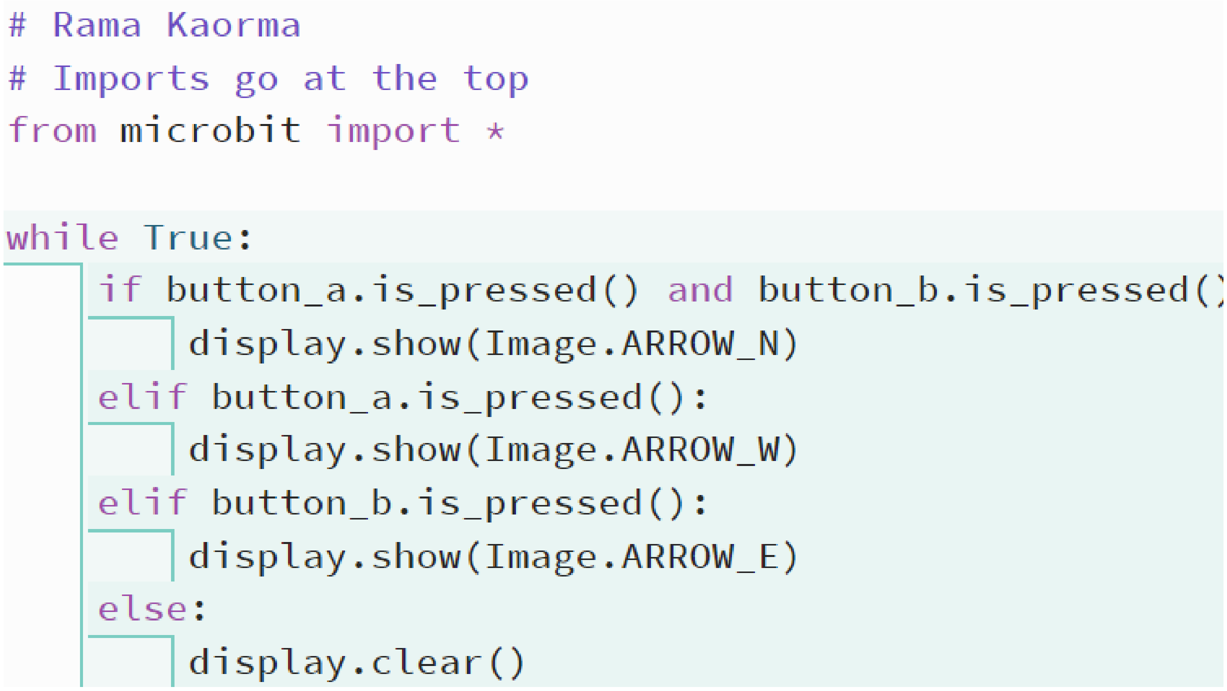
Write a program that points to button A when button A is pressed and to button B when button B is pressed. Otherwise, clear the display.



```
# Rama Kaorma
# Imports go at the top
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.ARROW_W)
    elif button_b.is_pressed():
        display.show(Image.ARROW_E)
    else:
        display.clear()
```

Task 3.3: CHALLENGE

Write a program that points to button A when button A is pressed and to button B when button B is pressed. **The program should point up if both buttons are pressed simultaneously.** Otherwise, clear the display.

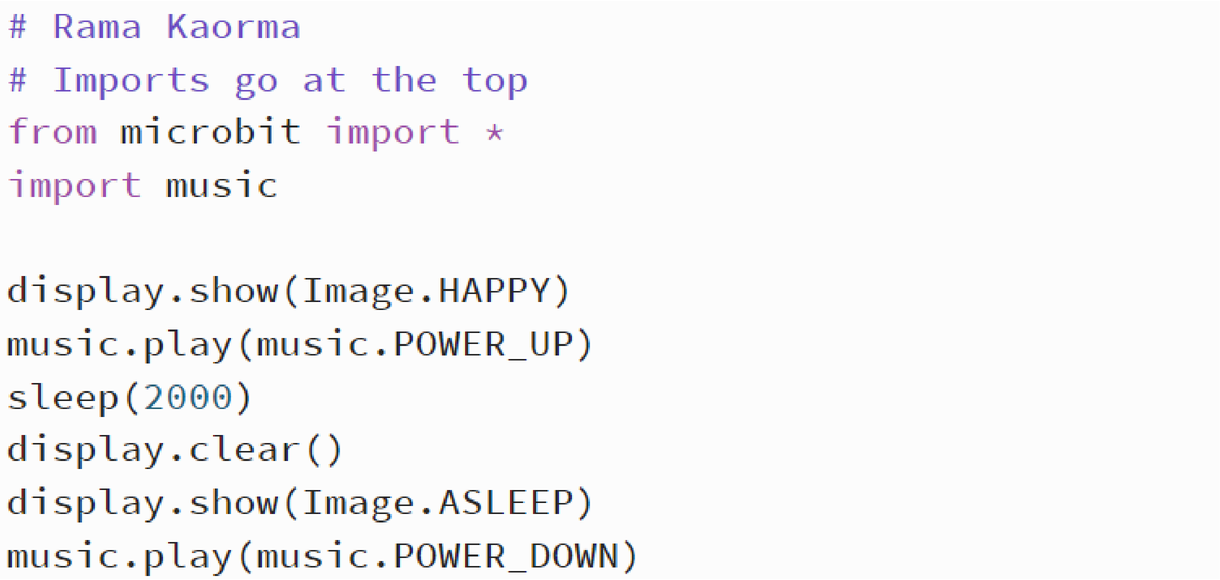


```
# Rama Kaorma
# Imports go at the top
from microbit import *

while True:
    if button_a.is_pressed() and button_b.is_pressed()
        display.show(Image.ARROW_N)
    elif button_a.is_pressed():
        display.show(Image.ARROW_W)
    elif button_b.is_pressed():
        display.show(Image.ARROW_E)
    else:
        display.clear()
```

Part 4: Music

Task 4.1: Power up, power down

Write a program that displays a smiley face and plays POWER_UP, pauses for 2 seconds, then displays the sleepy face and plays POWER_DOWN. Be careful of the order of operations, *and make sure the program is not running forever!* Note, the smiley face is HAPPY, sleepy face is ASLEEP.

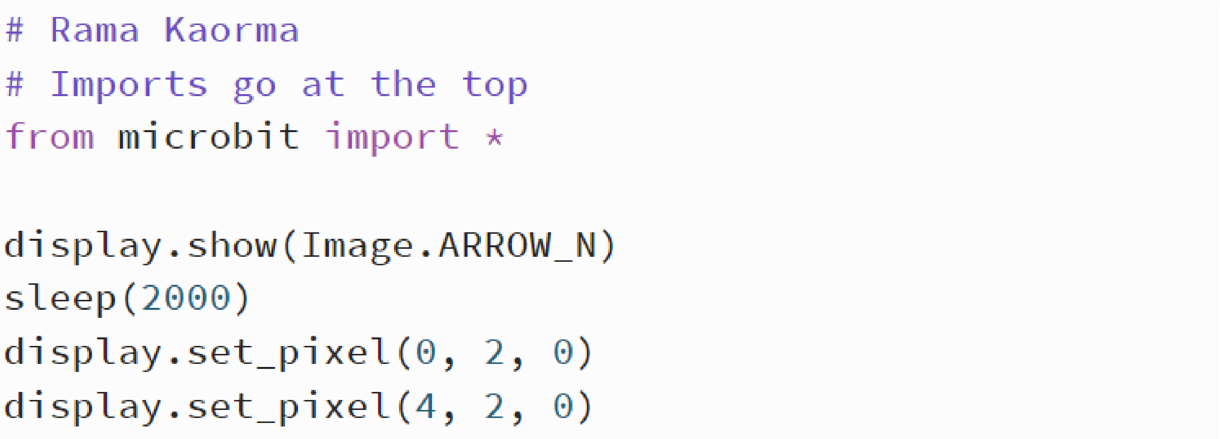


```
# Rama Kaorma
# Imports go at the top
from microbit import *
import music

display.show(Image.HAPPY)
music.play(music.POWER_UP)
sleep(2000)
display.clear()
display.show(Image.ASLEEP)
music.play(music.POWER_DOWN)
```

Part 5: One pixel at a time and for loops

Task 5.1: Shrink an arrow

Start with an arrow pointing north (upwards). You want your program to stay in that state for 2 seconds, then you want your arrow to shrink!

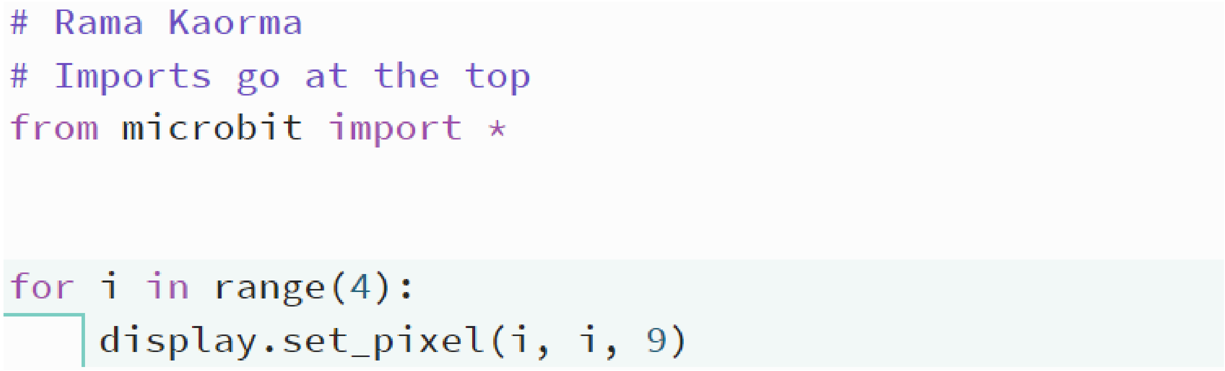


```
# Rama Kaorma
# Imports go at the top
from microbit import *

display.show(Image.ARROW_N)
sleep(2000)
display.set_pixel(0, 2, 0)
display.set_pixel(4, 2, 0)
```

Hint: How about you turn off some pixels?

Task 5.2: A more elegant star

Use a for loop to recreate the falling star problem. Notice that the y coordinate is not the only one repeating!



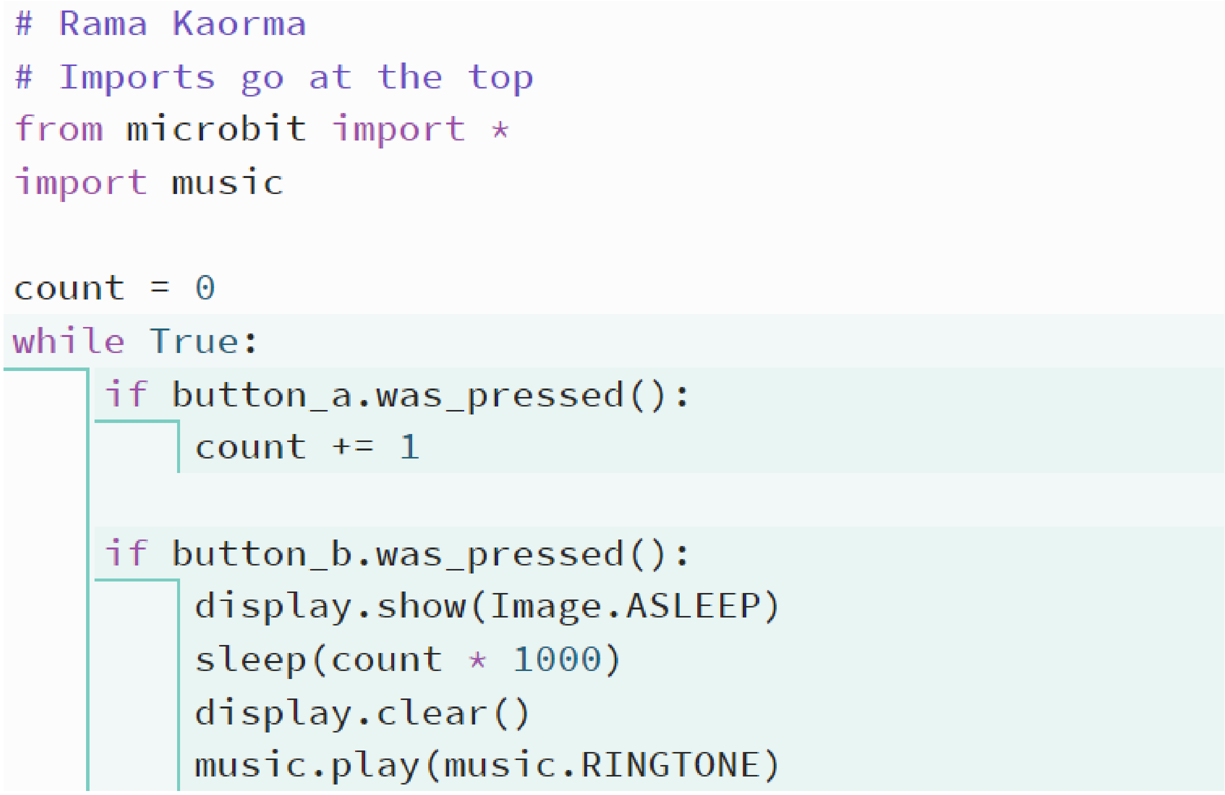
```
# Rama Kaorma
# Imports go at the top
from microbit import *


for i in range(4):
    display.set_pixel(i, i, 9)
```

Part 6: Counters and effects

Task 6.1: What we learned so far

Now, we’re going to write one big program with everything we have learned so far!
Write a program that uses Button A to set the timer in seconds. Your program should count the number of times that Button A was pressed.
For example, 3 presses on Button A sets the timer for 3 seconds.
When Button B is pressed:

- display Image.ASLEEP to show the timer has started;
- sleep for the time in seconds; then
- clear the screen and play music.RINGTONE.

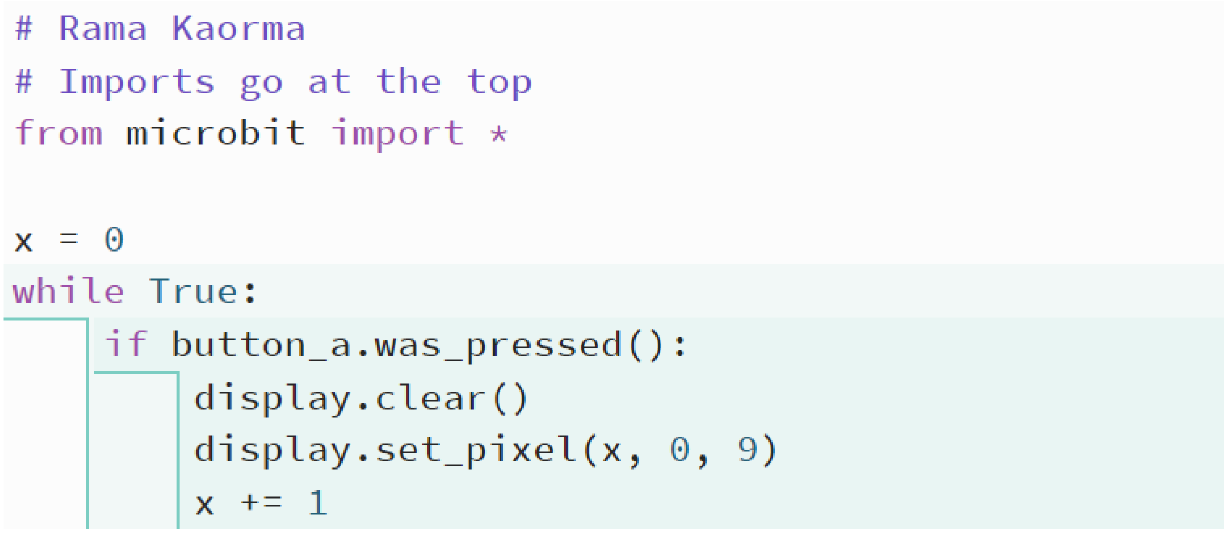


```
# Rama Kaorma
# Imports go at the top
from microbit import *
import music

count = 0
while True:
    if button_a.was_pressed():
        count += 1

    if button_b.was_pressed():
        display.show(Image.ASLEEP)
        sleep(count * 1000)
        display.clear()
        music.play(music.RINGTONE)
```

Part 7: Move!

Task 7.1: Move with buttons

Recreate the example from the slides where a dot moves horizontally. However, this time, we only want the dot to move when we press button A.

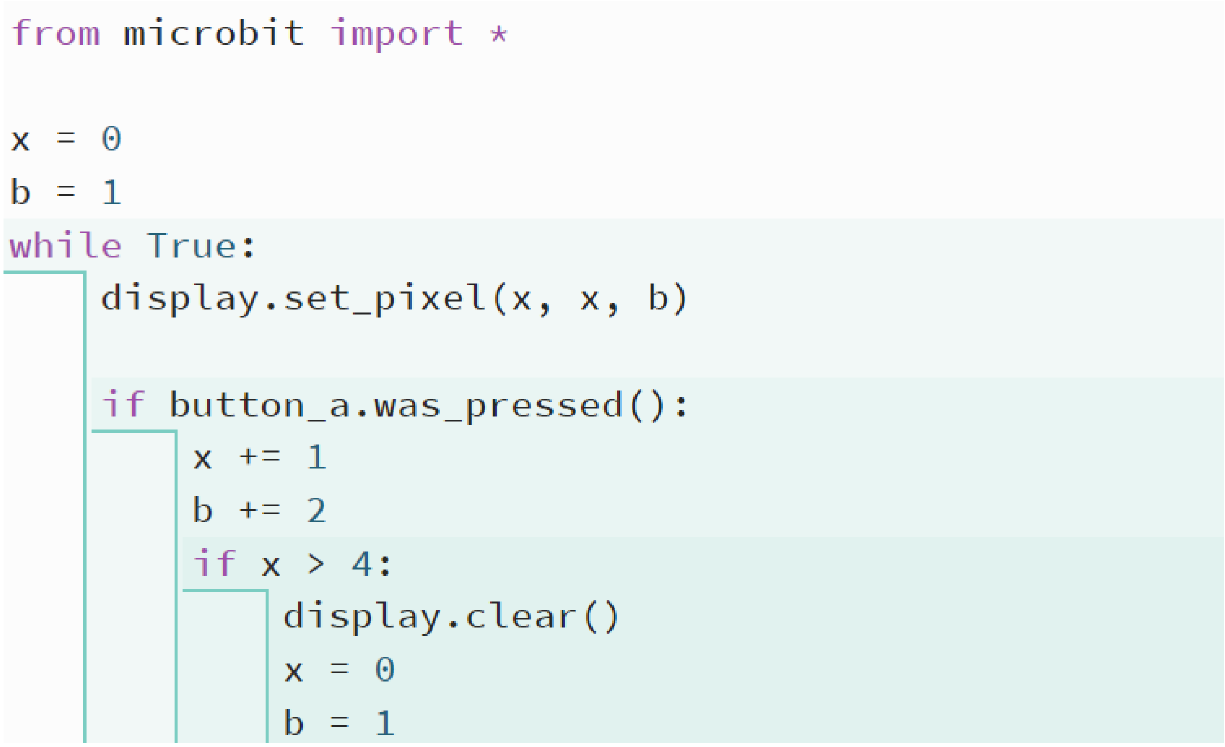


```
# Rama Kaorma
# Imports go at the top
from microbit import *

x = 0
while True:
    if button_a.was_pressed():
        display.clear()
        display.set_pixel(x, 0, 9)
        x += 1
```

Task 7.2: A falling star

Remember the falling star example? Write a program that imitates a falling star.

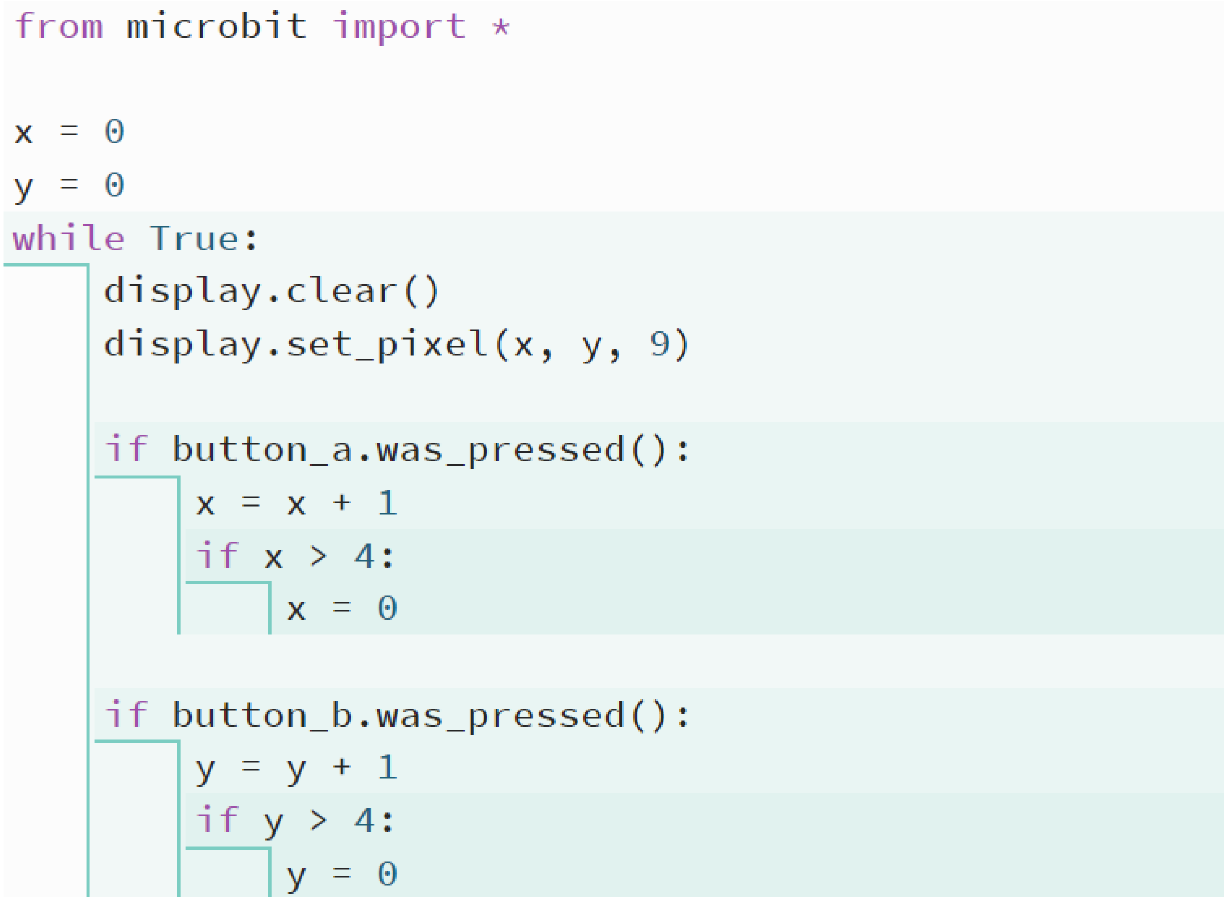
- Start with the corner pixel at brightness 1
- Every time you press button A, the next diagonal pixel lights up with a brightness 2 levels greater than the previous one.
- When you reach the final diagonal, the star should reset to only the corner pixel with brightness 1.



```python
from microbit import *

x = 0
b = 1
while True:
    display.set_pixel(x, x, b)

    if button_a.was_pressed():
        x += 1
        b += 2
        if x > 4:
            display.clear()
            x = 0
            b = 1
```

Task 7.3: Moving in 2D

Write a program that starts with a lit pixel at the coordinate (0,0). If button A is pressed, move the dot 1 pixel to the right. If button B is pressed, move the dot 1 pixel down. If x > 4 or y > 4, reset only that coordinate.



```
from microbit import *

x = 0
y = 0
while True:
    display.clear()
    display.set_pixel(x, y, 9)

    if button_a.was_pressed():
        x = x + 1
        if x > 4:
            x = 0

    if button_b.was_pressed():
        y = y + 1
        if y > 4:
            y = 0
```

Task 7.4: Snake!

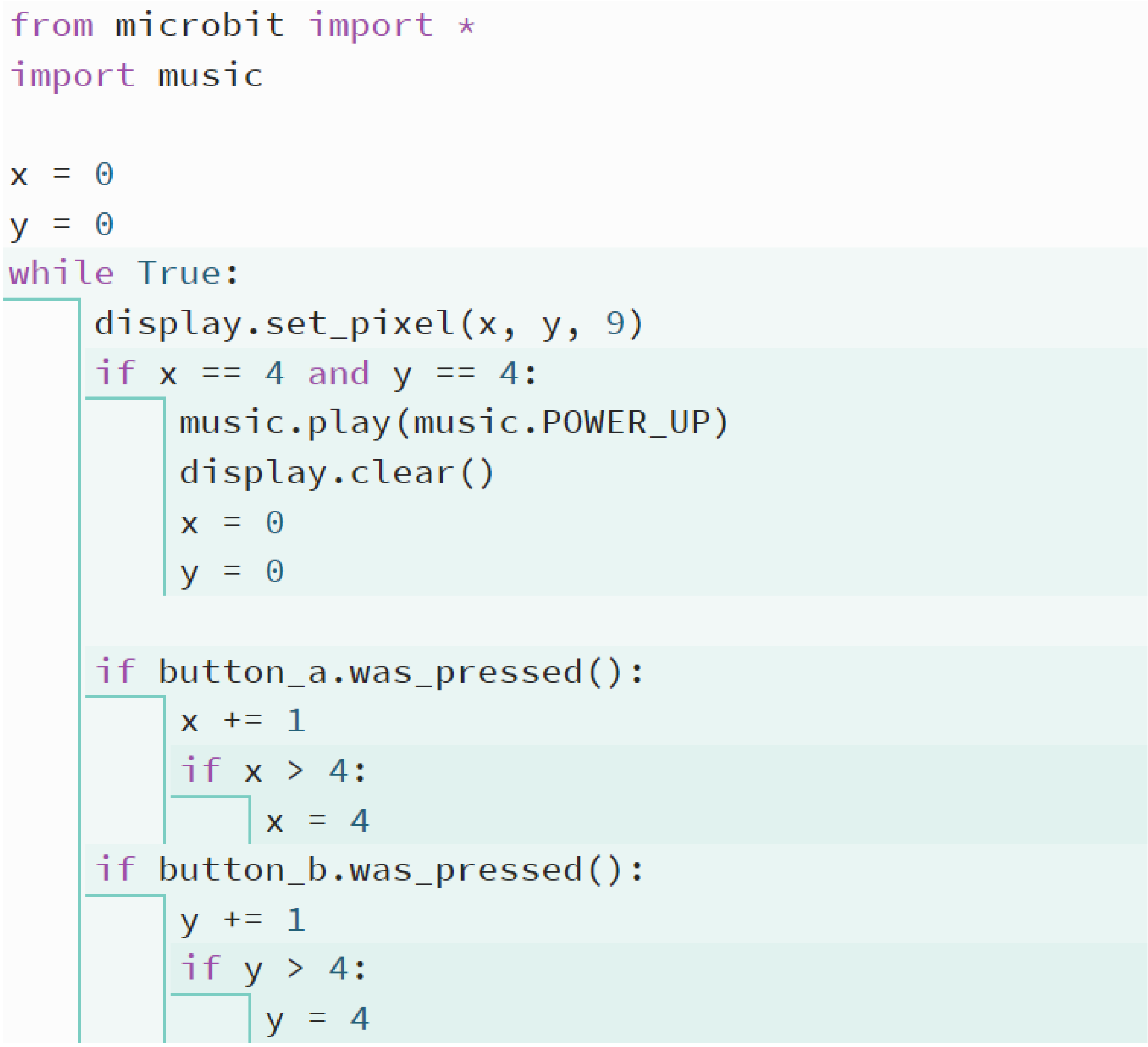
We now know enough to create a simple game of snake!

Our snake starts as one pixel in the top-left corner. The goal is to move it to the bottom-right corner using the buttons:

- Button A moves the snake to the right
- Button B moves the snake downwards

Grow the snake as you move it by not clearing the old pixels. Pressing a button to move the snake off the edges of the screen should have no effect.

When it reaches the bottom-right at (4, 4), celebrate by playing music.POWER_UP! Then reset the game by clearing the display and restarting the snake in the top-left corner.



```
from microbit import *
import music

x = 0
y = 0
while True:
    display.set_pixel(x, y, 9)
    if x == 4 and y == 4:
        music.play(music.POWER_UP)
        display.clear()
        x = 0
        y = 0

    if button_a.was_pressed():
        x += 1
        if x > 4:
            x = 4
    if button_b.was_pressed():
        y += 1
        if y > 4:
            y = 4
```

Let’s save:

Press on the 3 dots next to the save button. Click on ‘Save Python script’. This will download the code locally.