Welcome to CS 100A!

CA: Andrea Collins

Quick icebreakers ☃️

pollev.com/acollins010

Introduce yourself!

- Name
- Pronouns, if comfortable
- Class year
- Where is do you call home?
- 🌹 Rose, bud, thorn: something good that's happened recently, something bad that's happened recently, and something you're excited for

About me

- Bachelors in CS, currently doing a coterm masters degree in management science and engineering
- Home is Chicagoland area
- Took CS 106A my frosh fall and SLed CS 106A for three quarters
- I took ACE for CME 100 as a frosh and it saved my life 🙏

About this class 🧑‍🏫

Our main goals

- Learn fundamental CS skills: debugging, coding, pseudocoding, decomposition
- Learn how to navigate and use CS department resources to succeed
- Form a community within CS 106A
- Have fun!

Course logistics

- Section: **Thursdays 6:00-7:50 pm in McMurtry Art 360**
  - Section attendance is **required!**
  - we'll end exactly at 7:50, if you have extra questions please come before not after class
- 1:1 office hours: **Tuesdays 3-4pm, Wednesdays 4-5pm in Huang basement**
  - Room will hopefully change next week
  - Sign up here
  - Conceptual questions only; I can't look at your homework code :(
- Group office hours: **Tuesdays 4-5pm in Huang basement**
  - Conceptual questions only; I can't look at your homework code :(
- Midterm/final review parties: dates TBD, hopefully we'll have food! 🍕
- All communication over **Slack!** Please join if you haven't yet
  - email me if you didn't receive an invitation to join :)
- Course enrollment: codes sent out this week, enroll on Axess

Support systems for CS 106A

Where can I find support in CS 106A?

- LaIR
  - homework, conceptual, and debugging questions
  - logistics: 3-4th floors of Durand, 7-11pm PST Sunday-Thursday, sign up online
  - more details on exactly how to get there here
  - you don't need a specific question to sign up for LaIR! Even if you're just stuck, the SLs would love to help you
- Nick Parlante's office hours
  - Nick is a pretty friendly guy! I challenge everyone to go to his OH once this quarter.
  - CS career and academic advice, conceptual questions, questions from lecture
- Elyse's office hours
  - conceptual questions, homework questions, CS career and academic advice, questions from lecture
- Andrea's office hours
  - conceptual questions, CS career and academic advice, need help prepping to go to other OH, questions from ACE section
- CS 106A Ed forum
  - anonymous questions on homework, CS 106A section, lecture, course logistics

You don't need to have a specific question in mind to go to office hours!

Tips for success in CS 106A

- Seek help sooner rather than later
  - I went to LaIR for every single assignment in CS 106A. No shame.
  - If you don't know where to seek help, ask me!
- Start the assignments early
  - You will almost never write perfect code on the first try. Debugging takes time, so leave yourself lots of time to debug before the due date.
- Emergencies will come up, and that's okay
  - Let me and Elyse know how we can support you when unforeseen circumstances come up.
- Ask questions!
  - Especially here in ACE, we're all in this together.
  - Basically the whole goal of section is to get students to ask questions, so please help me out :)

Now, onto a conceptual review…

Turn to a partner.

👍

What's a variable?
Give an example.

♻️

What are some differences between for- and while- loops?

🧑‍🔧

What's a function? How do you define a function in Python?

Variables

```
is_ACE_cool = True

time = 6 + 1

my_name = 'Andrea'
```

All these are variables!

`time` is now a variable that refers to the value `7`.

```
time = time + 1
```

Now `time` is a variable that refers to the value `8`.

Which of these examples are **not** good variable names? Why?

```
x = 1

book_title = "Green Eggs and Ham"

number = 90

is_true = True

number_of_students = 24
```

Which of these examples are **not** good variable names? Why?

```
x = 1

book_title = "Green Eggs and Ham"

number = 90

is_true = True

number_of_students = 24
```

`Variable names should be descriptive and not too long!`

Loops

Anatomy of a while loop

```
number = 0

while number < 7:

    number = number + 1
```

What will the variable `number` equal after the while loop is done running?

Anatomy of a while loop

```
number = 0

while number < 7:

    number = number + 1
```

What will the variable `number` equal after the while loop is done running?

6

Anatomy of a while loop

```
number = 0

while number < 7:

    number = number + 1
```

How many times will this loop run?

6 times

Anatomy of a while loop

```
number = 0

while number < 7:

    number = number + 1
```

Loop runs 6 times -> loop runs for 6 **iterations**

Anatomy of a while loop

```
number = 0

while number < 7:

    number = number + 1
```

loop condition: any code inside the while loop will execute WHILE this statement is true.

Anatomy of a while loop

```
number = 0

while number < 7:

    number = number + 1
```

loop code: this code will execute every time the loop runs (once every iteration)

Anatomy of a for loop

```
image = SimpleImage(filename)

image_width = image.width

for pixel in image:

    pixel.green = 0

    pixel.blue = 0

    pixel.red = 255
```

What will the image look like after the for loop is done running?

Anatomy of a for loop

```
image = SimpleImage(filename)

image_width = image.width

for pixel in image:

    pixel.green = 0

    pixel.blue = 0

    pixel.red = 255
```

How many times will this loop run?

Anatomy of a for loop

```
image = SimpleImage(filename)

image_width = image.width

for pixel in image:

    pixel.green = 0

    pixel.blue = 0

    pixel.red = 255
```

the variable: inside the collection, we want to iterate over each variable

the collection: contains the individual variables we want to iterate over

Anatomy of a for loop

```
image = SimpleImage(filename)
image_width = image.width
for pixel in image:
    pixel.green = 0
    pixel.blue = 0
    pixel.red = 255
```

all the code inside the loop will execute once for each variable in the collection (for each iteration)

Anatomy of a for loop

```
image = SimpleImage(filename)

image_width = image.width

for pixel in image:

    pixel.green = 0

    pixel.blue = 0

    pixel.red = 255
```

number of iterations of a for loop = number of variables in collection

Range-based for loops

- uses `range()` function to loop over a collection of numbers
- use range-based for loop when
  - you need to know what iteration number you're on
  - example: you only want to iterate over a few variables in a collection
- use regular for loop when
  - you don't need to know what iteration number you're on
  - example: you want to iterate over every single variable in a collection

Anatomy of a range-based for loop

```
image = SimpleImage(filename)

image_width = image.width

for i in range(image_width):

   print(i)
```

Anatomy of a range-based for loop

```
image = SimpleImage(filename)

image_width = image.width

for i in range(image_width):
    print(i)
```

`range(number)` creates a collection of numbers from 0 to number, exclusive of number

do something with the variable `i`

Anatomy of a range-based for loop

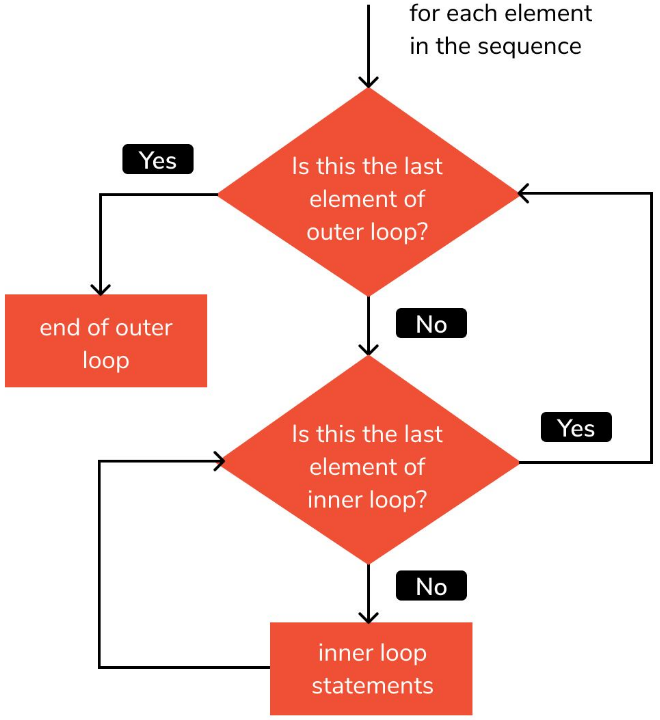
```
image = SimpleImage(filename)

image_width = image.width

for i in range(image_width):

    print(i)
```

How many iterations will this loop run for?

Nested for loops

- for loop within a for loop…inception!
- note: you can also nest while loops and for loops in a very similar manner, but the nested for loop structure is super common so we'll dive deeper into it today

How do I read nested for loops?

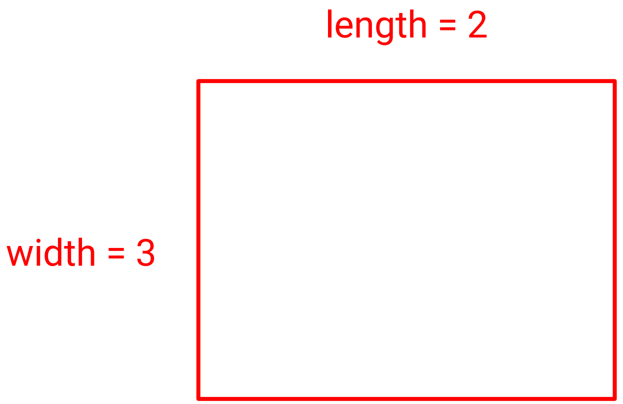

for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

For how many iterations will this nested for loop run?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j)
```

For how many iterations will this nested for loop run?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j)
```



Kind of like the area of a box!
3 outer loop iterations x 2 inner loop iterations = 6 total iterations

What will this nested for loop print?

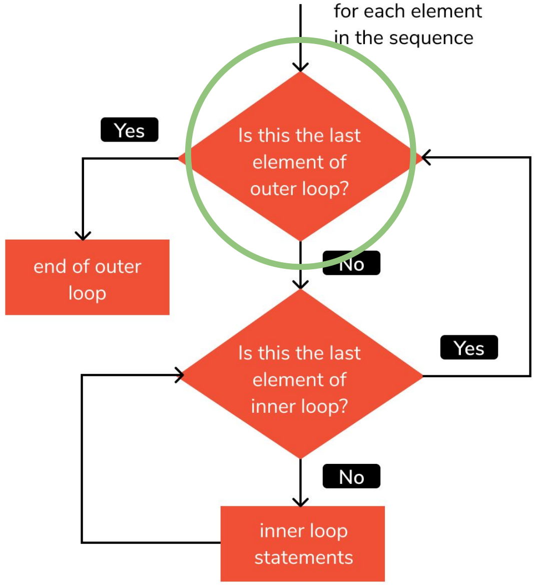
```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j)
```

Tip: computer science folks really like to use i and j as variable names for loop counters…I do not know why

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

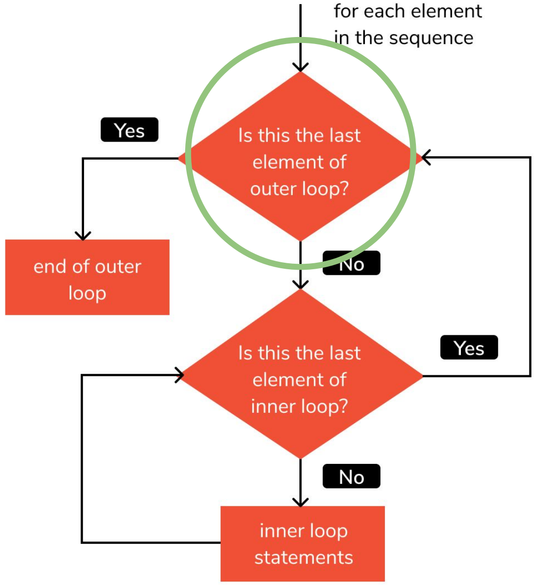
We will print:


for each element in the sequence
Is this the last element of outer loop?
Yes
No
end of outer loop
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

i will range from 0 to 3, exclusive

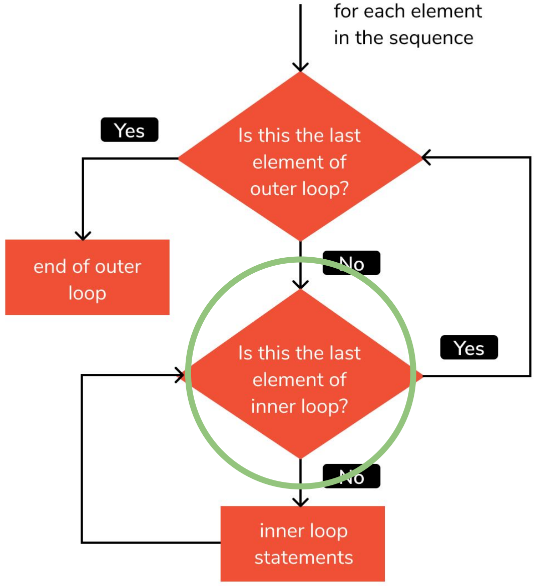

for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

We will print:

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

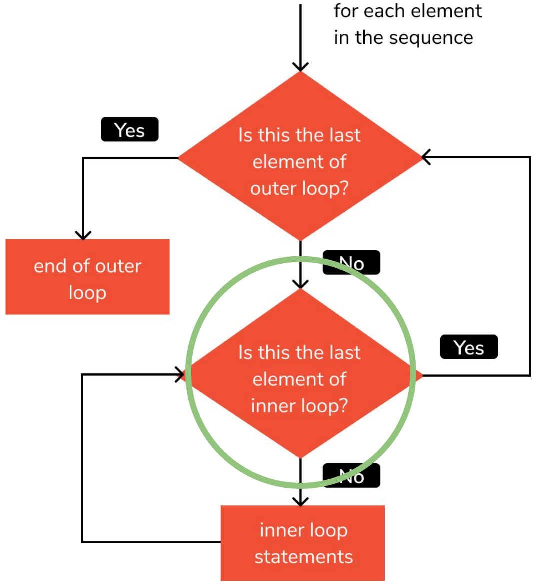
We will print:


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

j will range from 0 to 2, exclusive

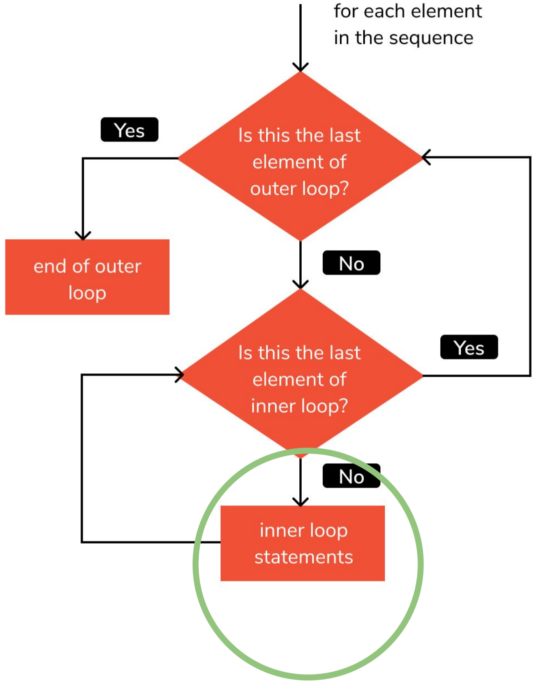

for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

We will print:

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

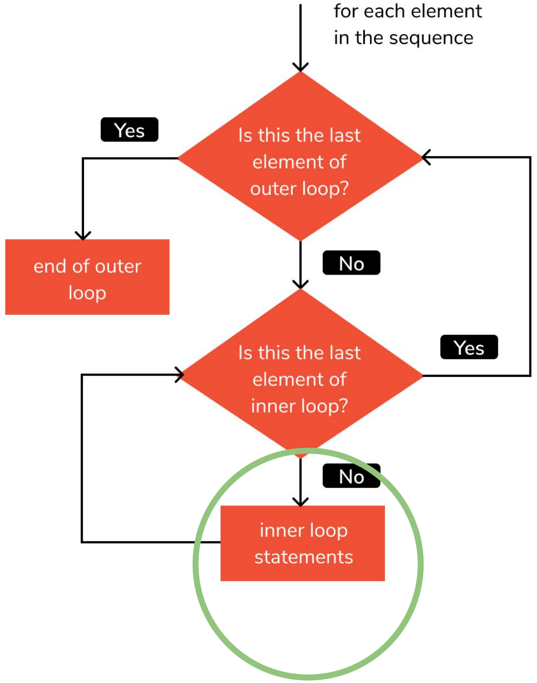
We will print:


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

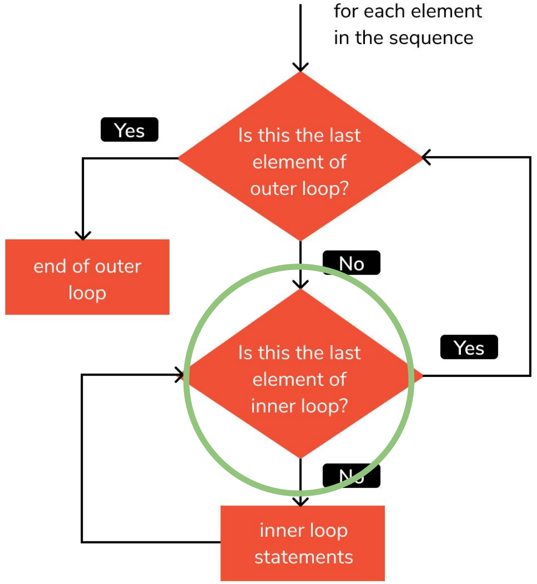
We will print: 0 0 |


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

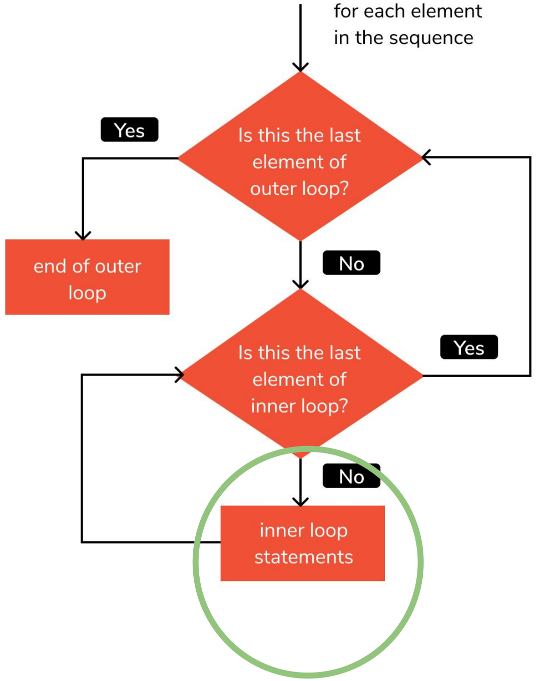
We will print: 0 0 |


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

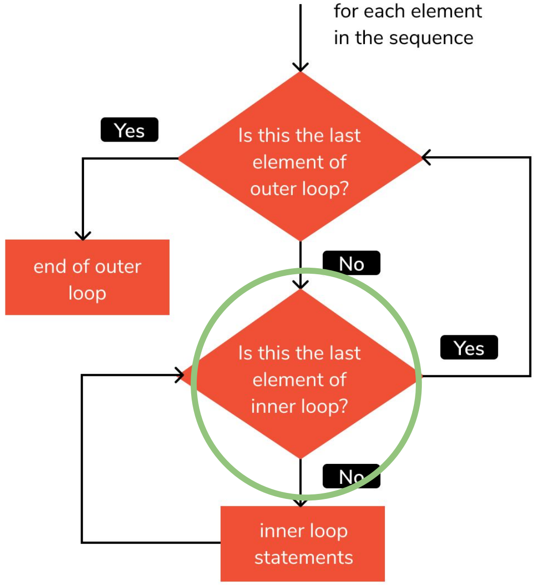
We will print: 0 0 | 0 1 |


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

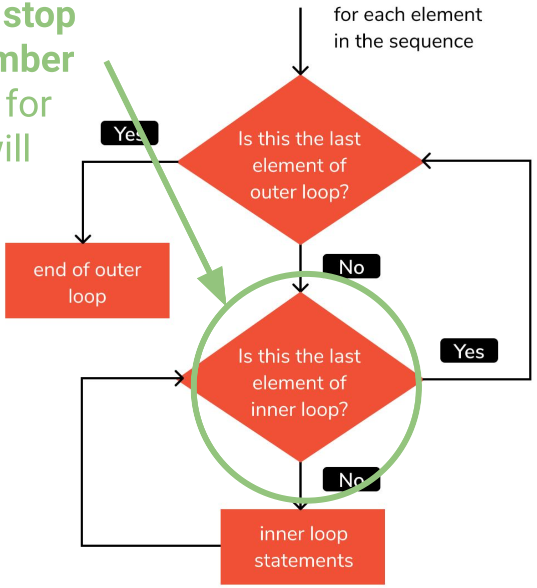
We will print: 0 0 | 0 1 |



What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

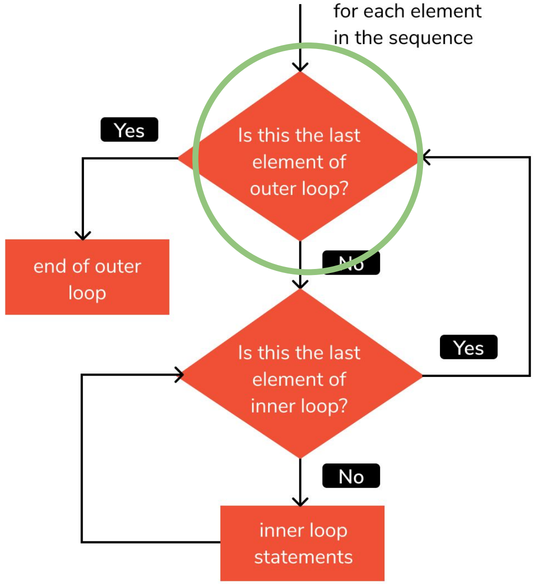
YES! The inner loop ranges from **0 to 2, exclusive**. That means that **2** is the **first number to stop** at, meaning **1 is the last number** that j is equal to in the inner for loop. So the inner for loop will evaluate exactly **2 times**.


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

We will print: 0 0 | 0 1 |

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

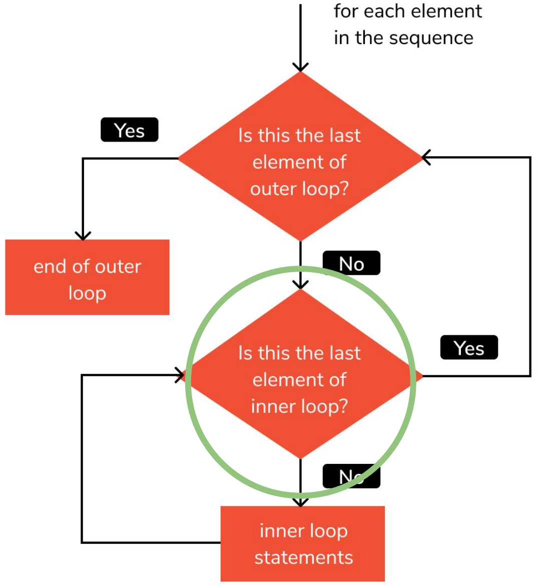

for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

We will print: 0 0 | 0 1 |

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

We will print: 0 0 | 0 1 |

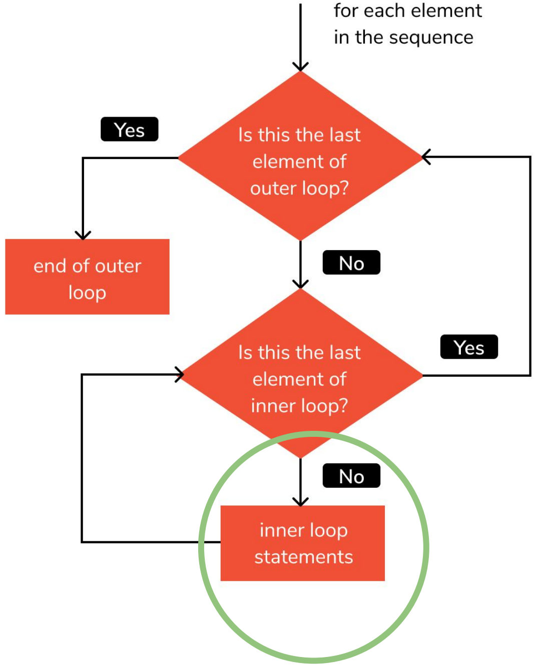


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

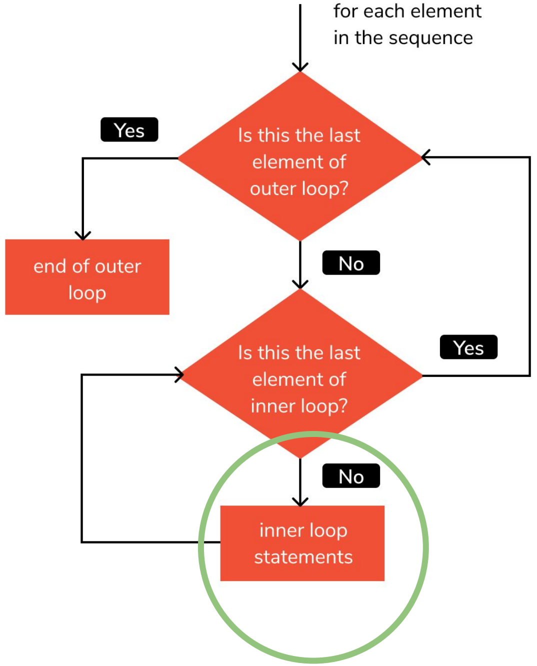
We will print: 0 0 | 0 1 |


for each element in the sequence
Is this the last element of outer loop?
Yes
No
end of outer loop
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

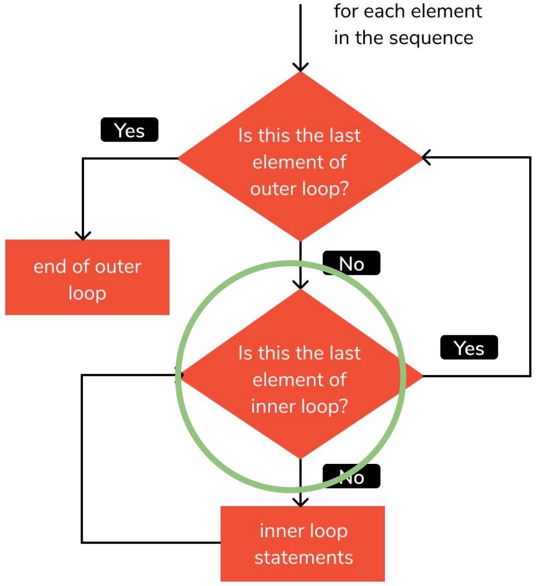
We will print: 0 0 | 0 1 | 1 0 |


for each element in the sequence
Is this the last element of outer loop?
Yes
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

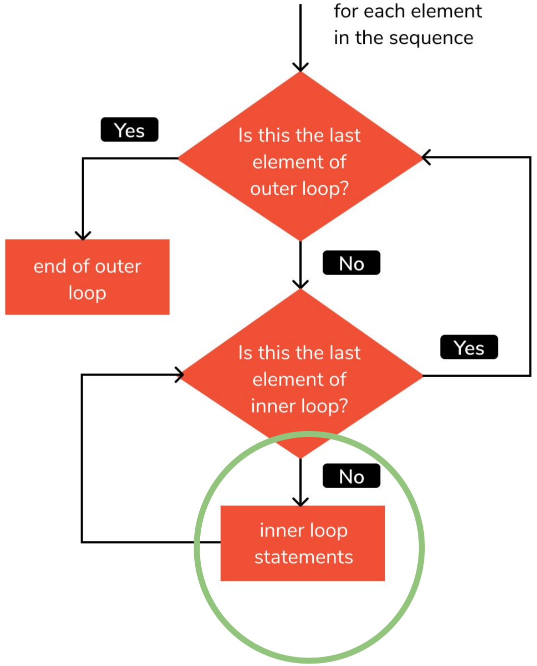
We will print: 0 0 | 0 1 | 1 0 |


for each element in the sequence
Is this the last element of outer loop?
Yes
No
end of outer loop
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

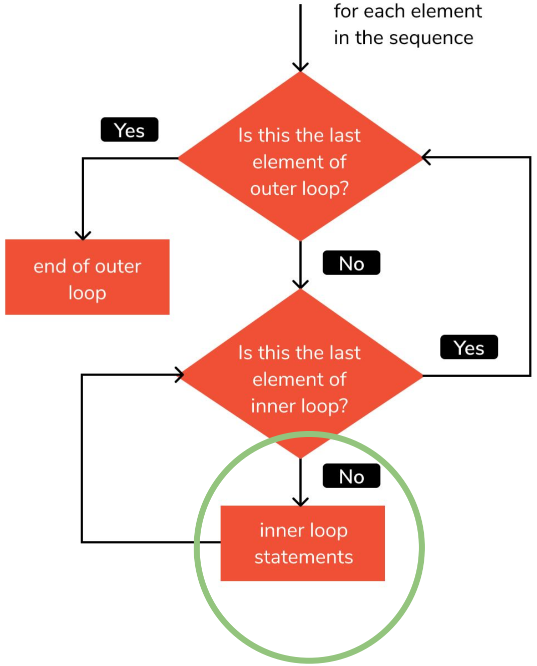
We will print: 0 0 | 0 1 | 1 0 |


for each element in the sequence
Is this the last element of outer loop?
Yes
No
end of outer loop
Is this the last element of inner loop?
Yes
No
inner loop statements

What will this nested for loop print?

```
width = 3
length = 2
for i in range(width):
    for j in range(length):
        print(i, j, '|')
```

We will print: 0 0 | 0 1 | 1 0 | 1 1 |



for each element in the sequence
Yes
Is this the last element of outer loop?
end of outer loop
No
Is this the last element of inner loop?
Yes
No
inner loop statements

and so on and so forth…

What will this nested for loop print?

```
width = 3

length = 2

for i in range(width):

    for j in range(length):

        print(i, j, '|')
```

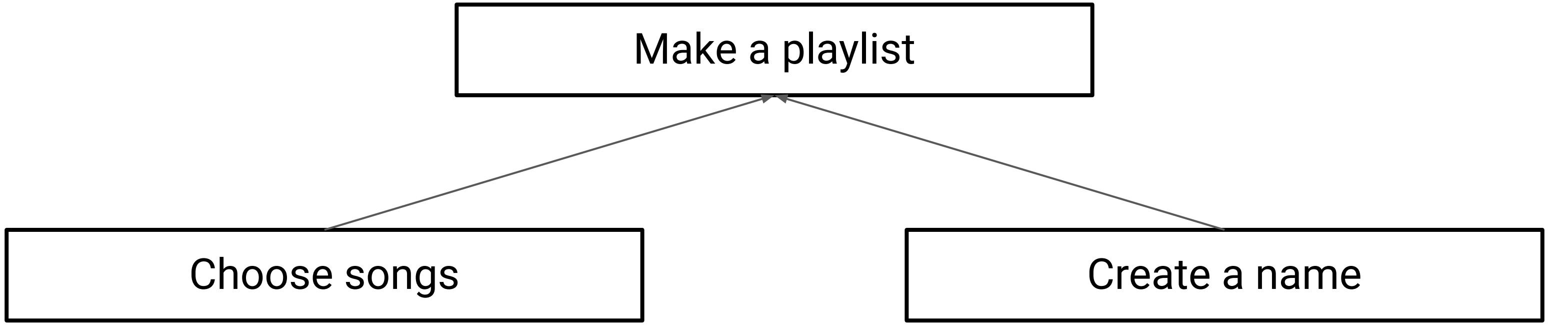
We will print: 0 0 | 0 1 | 1 0 | 1 1 | 2 0 | 2 1 |

Decomposition 🎶

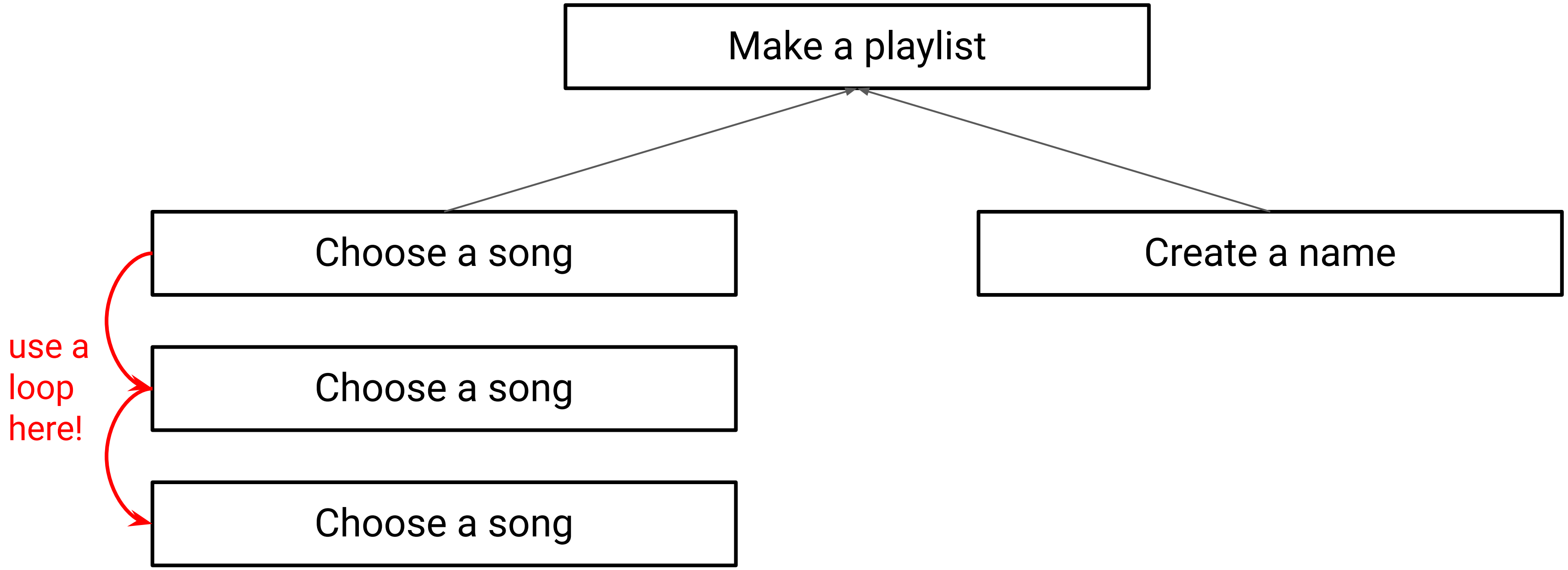
How can we break up a function into smaller parts?

Make a playlist

How can we break up a function into smaller parts?



How can we break up a function into smaller parts?



How can we decompose this function?

```
def makeImageRed(filename):
    image = SimpleImage(filename)
    image_width = image.width
    for pixel in image:
        pixel.green = 0
        pixel.blue = 0
        pixel.red = 255
```

How can we decompose this function?

```
def makeImageRed(filename):
    image = SimpleImage(filename)
    image_width = image.width
    for pixel in image:
        pixel.green = 0
        pixel.blue = 0
       pixel.red = 255
```

Hint: what exactly is this part of the function doing?

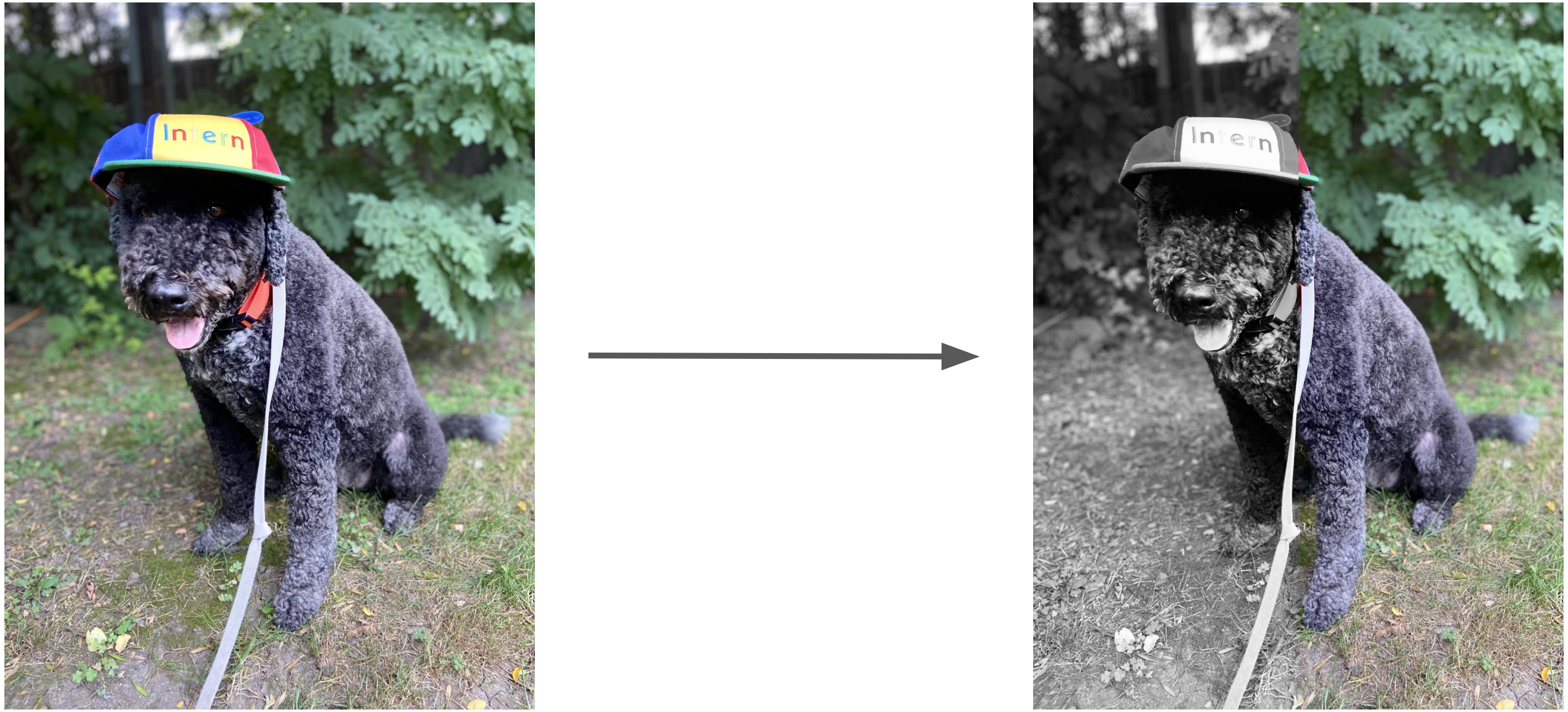
How can we decompose this function?

```
def makeImageRed(filename):
    image = SimpleImage(filename)
    image_width = image.width
    for pixel in image:
        makePixelRed(pixel)
```

```
def makePixelRed(pixel):
    pixel.green = 0
    pixel.blue = 0
    pixel.red = 255
```

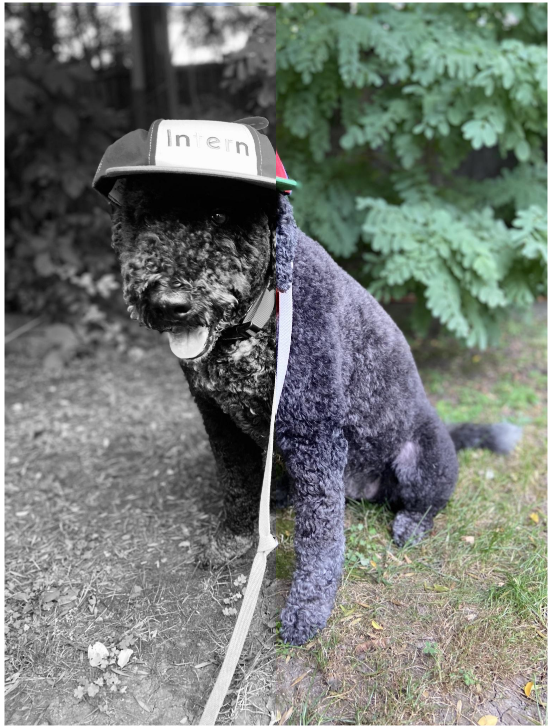
Let's try some image problems! 🤳🌉

1. You're implementing a cool new Instagram filter to turn half of an image "greyscale".



1. You're implementing a cool new Instagram filter to turn half of an image "greyscale".

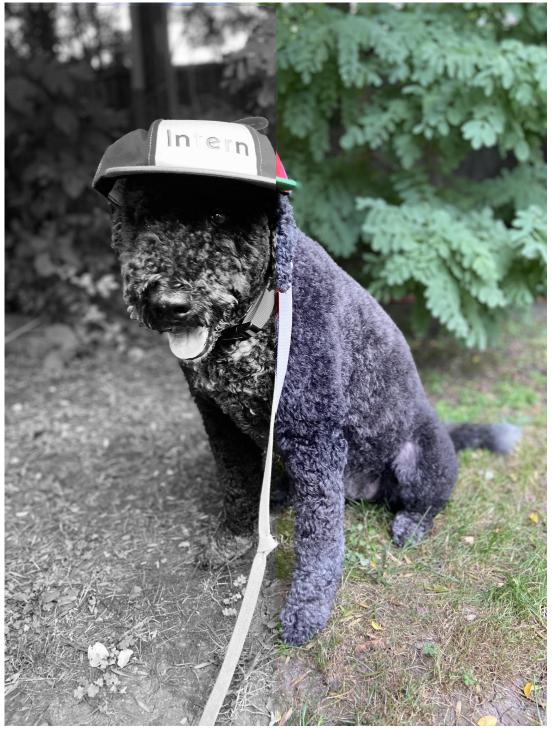
"greyscale": all the colors are black/white/grey



1. You're implementing a cool new Instagram filter to turn half of an image "greyscale".

Turns out we can turn a single pixel greyscale by setting each RGB value to the average of all the RGB pixel values!

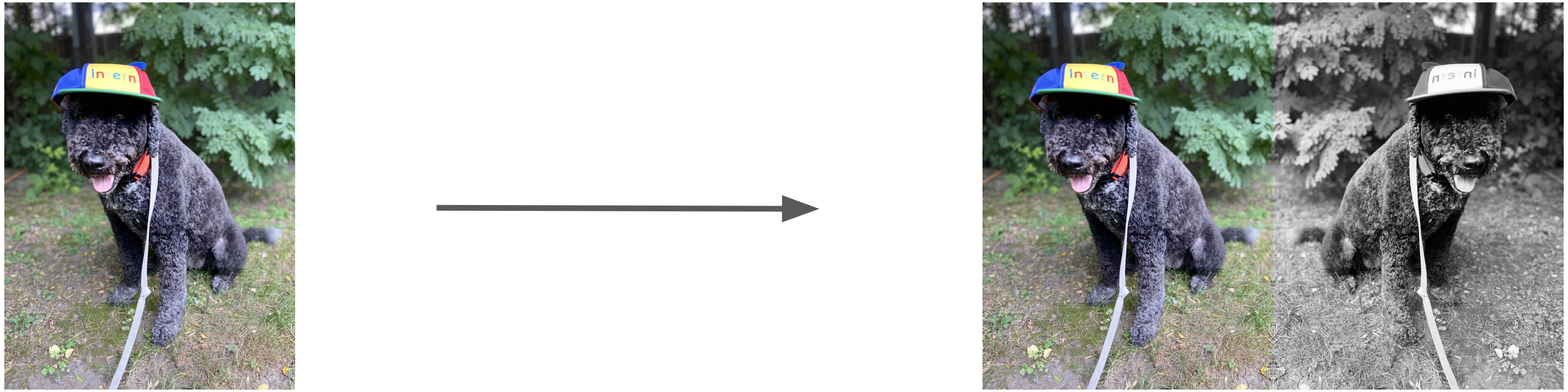
greyscale value = average(pixel red, pixel green, pixel blue)



Half greyscale solution

```
def greyscale_half_image(filename):
    image = SimpleImage(filename)
    for x in range(image.width):
        for y in range(image.height):
            pixel = image.get_pixel(x, y)
            greyscale_pixel(pixel)
def greyscale_pixel(pixel):
    avg = (pixel.red + pixel.green + pixel.blue) / 3
    pixel.red = avg
    pixel.green = avg
    pixel.blue = avg
```

2. You want to reflect an image over its y-axis and make the reflection greyscale.



Form a group of 2-3 and discuss for 10 min!

Solution

```
def reflect_and_greyscale(filename):
    image = SimpleImage(filename)
    new_image = SimpleImage.blank(image.width * 2, image.height)
    for x in range(image.width):
        for y in range(image.height):
            pixel = image.get_pixel(x, y)
            pixel_left_regular = new_image.get_pixel(x, y)
            pixel_right_reflected = new_image.get_pixel(out.width - 1 - x, y)
            copy_pixel(pixel, pixel_left_regular)
            copy_pixel(pixel, pixel_right_reflected)
            greyscale_pixel(pixel_right_reflected)
```

```
def copy_pixel(old_pixel, new_pixel):
    new_pixel.red = old_pixel.red
    new_pixel.green = old_pixel.green
    new_pixel.blue = old_pixel.blue

def greyscale_pixel(pixel):
    avg = (pixel.red + pixel.green +
    pixel.blue) / 3
    pixel.red = avg
    pixel.green = avg
    pixel.blue = avg
```

See you next week! 🥰