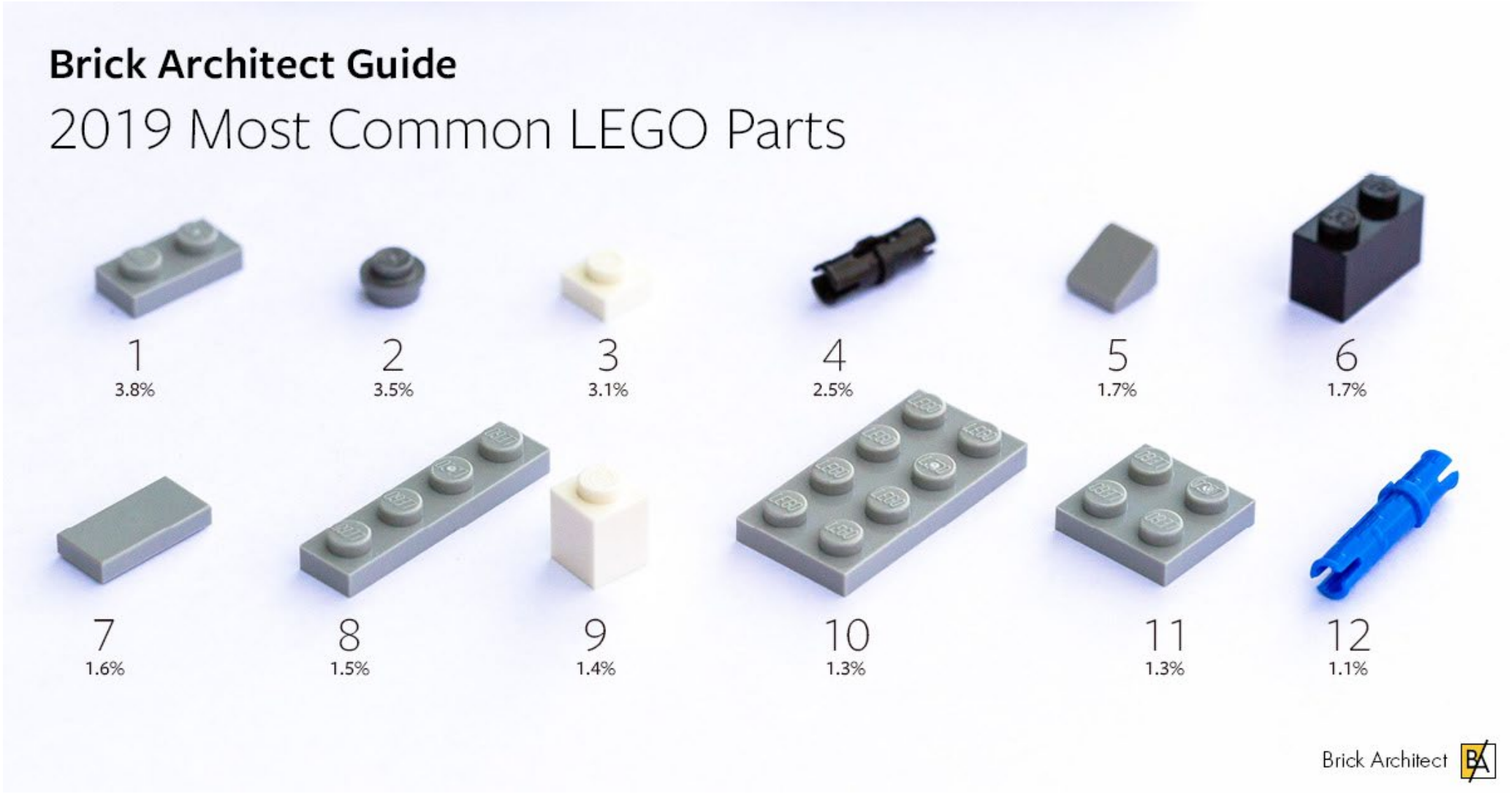
Survey: which LEGO block is the most painful to step on



COMP1511 Week 2!

T14A: 2pm – 5pm

Tutors: Me + Vivian Zheng

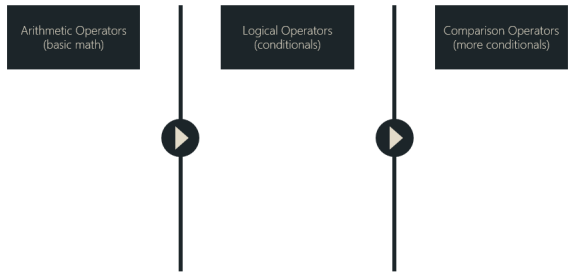
My GitHub:

https://github.com/william-o-s/unsw_comp1511_tutoring

The Agenda

Calculating Values (5 mins)

In groups, recall the operators for your category...



Sample Arithmetic (10 mins)

Recall the variable types we've seen so far...

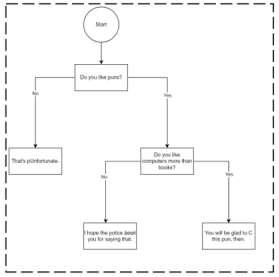
```
double
```

```
int
```

```
char
```

Diagramming (20 mins)

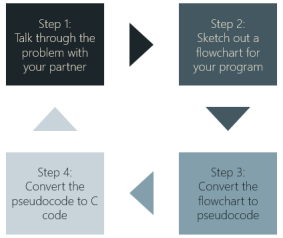
Let's practice diagramming our thought processes!



https://app.diagrams.net/

Coding Exercise (20 mins)

In pairs, attempt the programming exercise



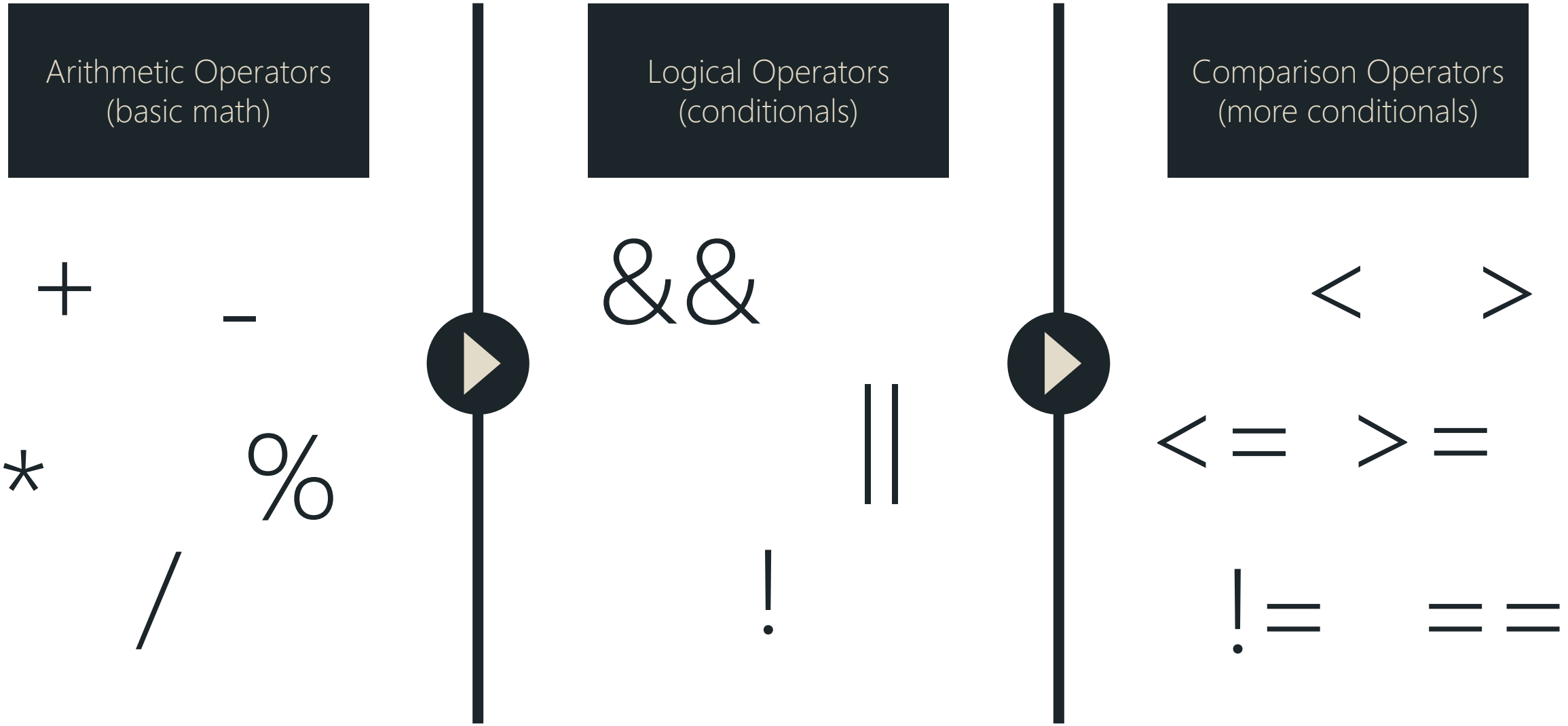
In groups, recall the operators for your category…

Arithmetic Operators
(basic math)

Logical Operators
(conditionals)

Comparison Operators
(more conditionals)

...did you get them all?



Oh, and what's the difference between these two?

/ vs. %

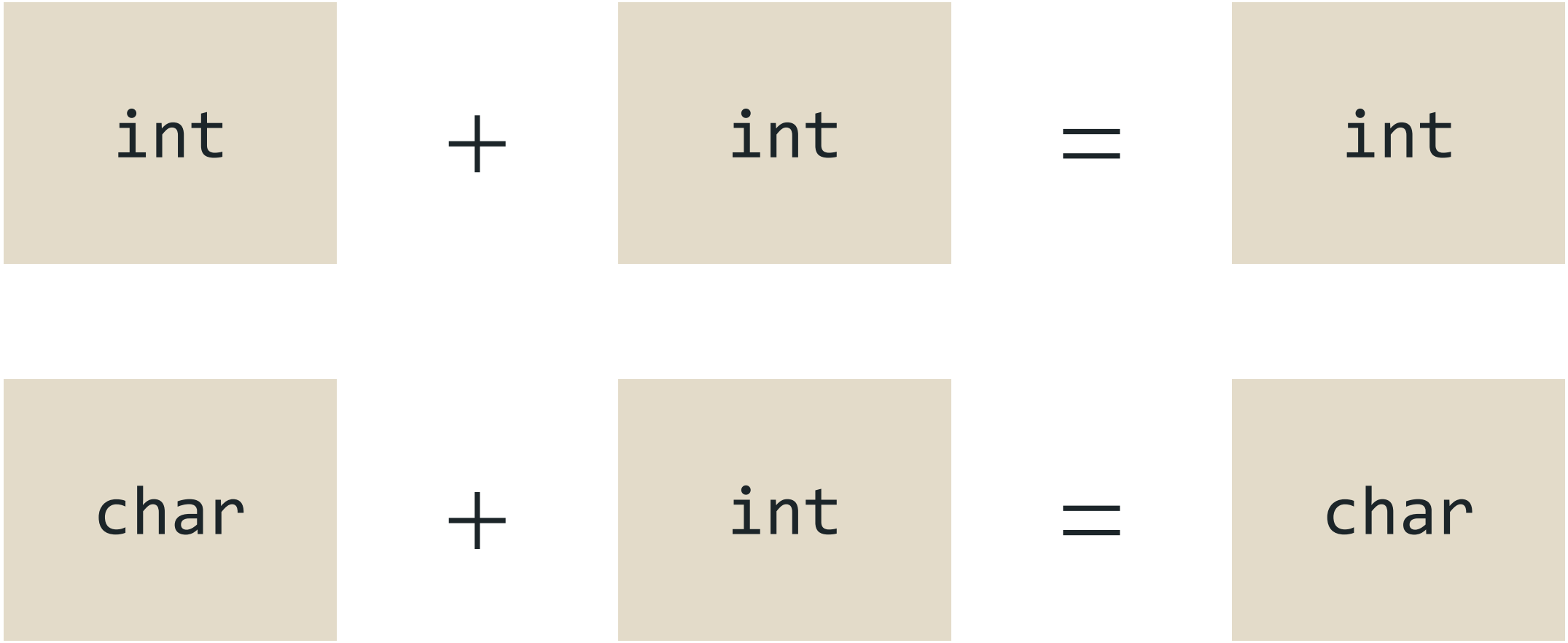
Recall the variable types we've seen so far…

`double`

`int`

`char`

...and recall the values of certain expressions...



…so let’s practice with some possible calculations

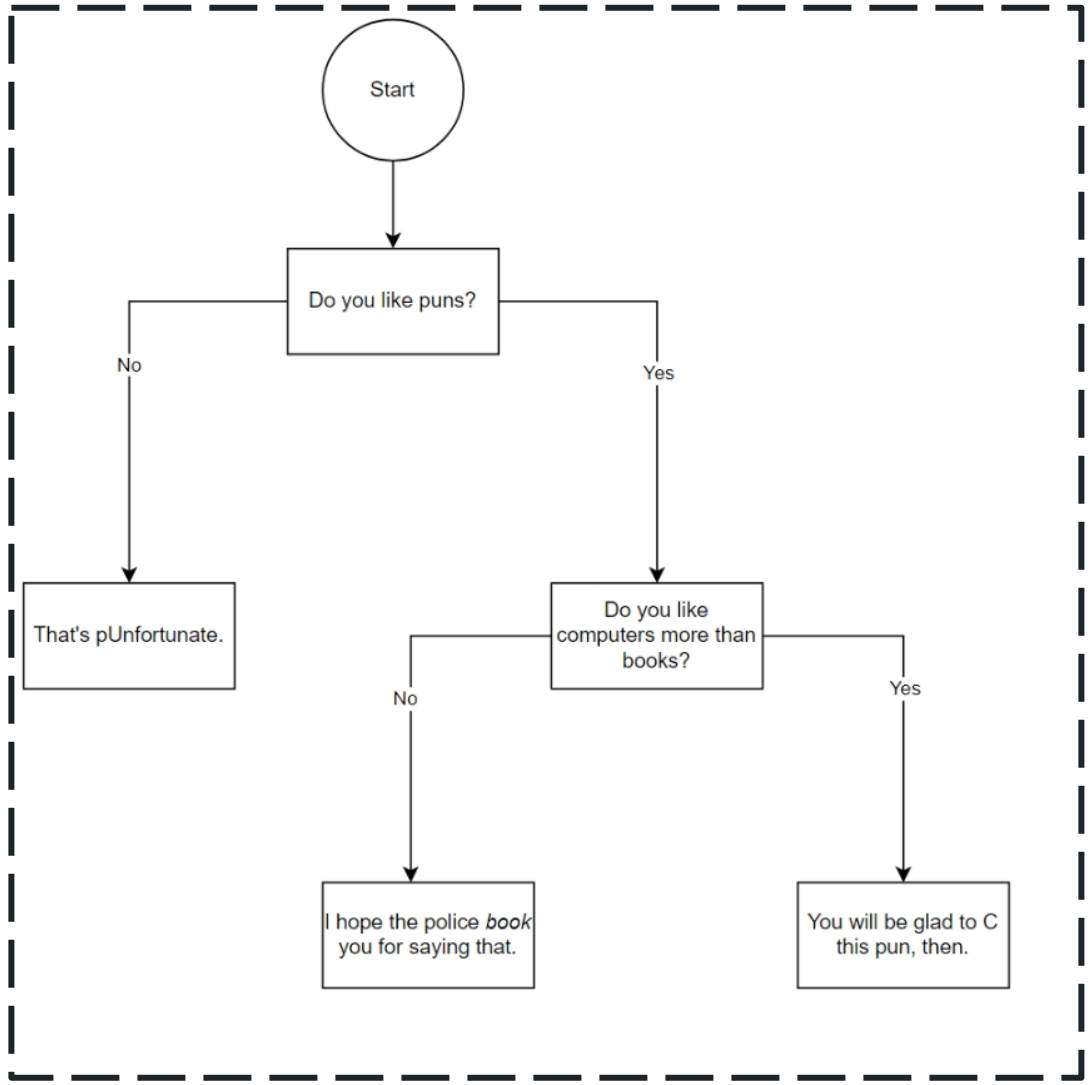
```
7 / 2
```
= ?

```
(3.0 / 2) + 1
```
= ?

```
‘a’ + 5
```
= ?

```
‘F’ – ‘A’ + ‘a’
```
= ?

Let’s practice diagramming our thought processes!



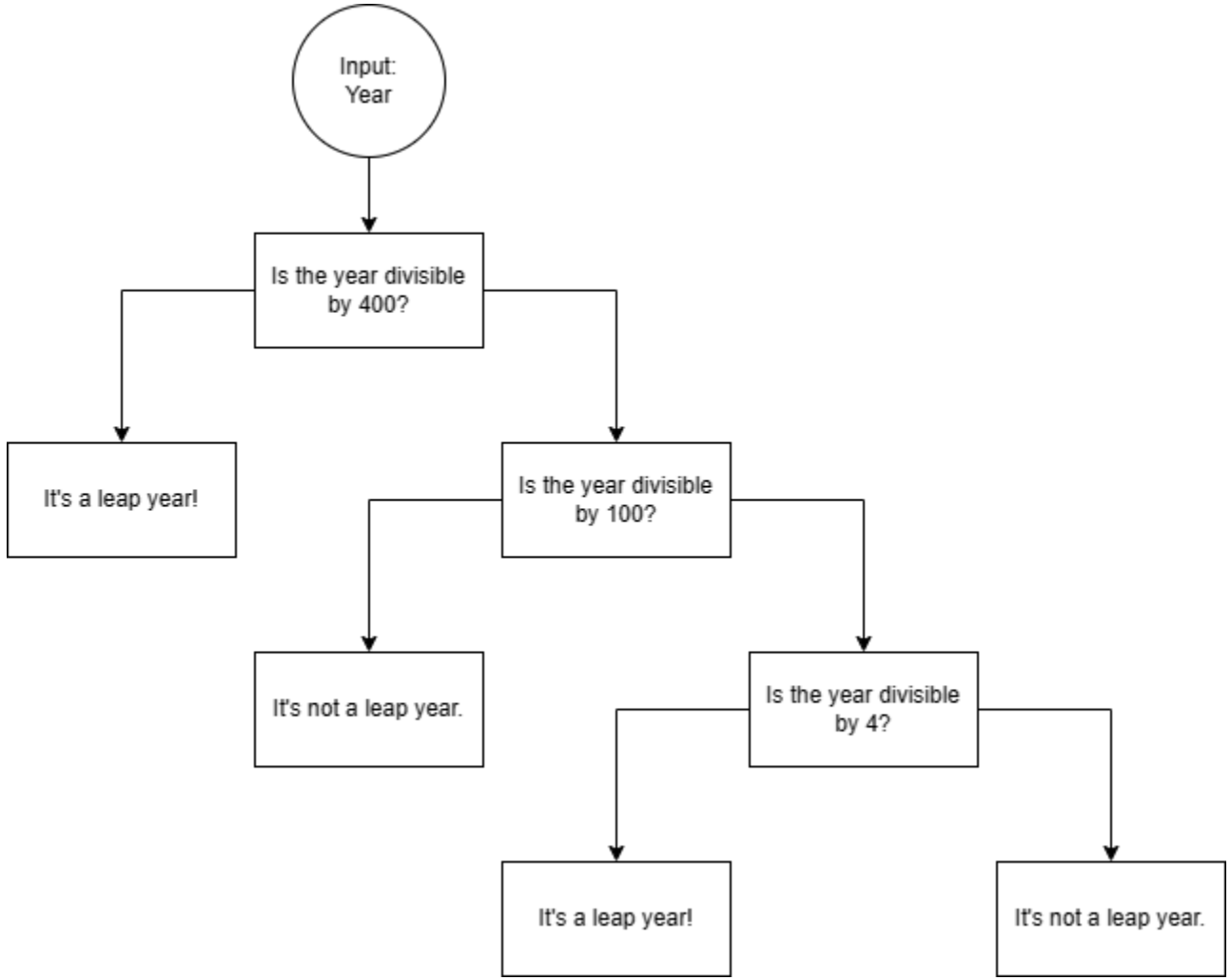
https://app.diagrams.net/

Let's try making a flowchart for determining leap years…

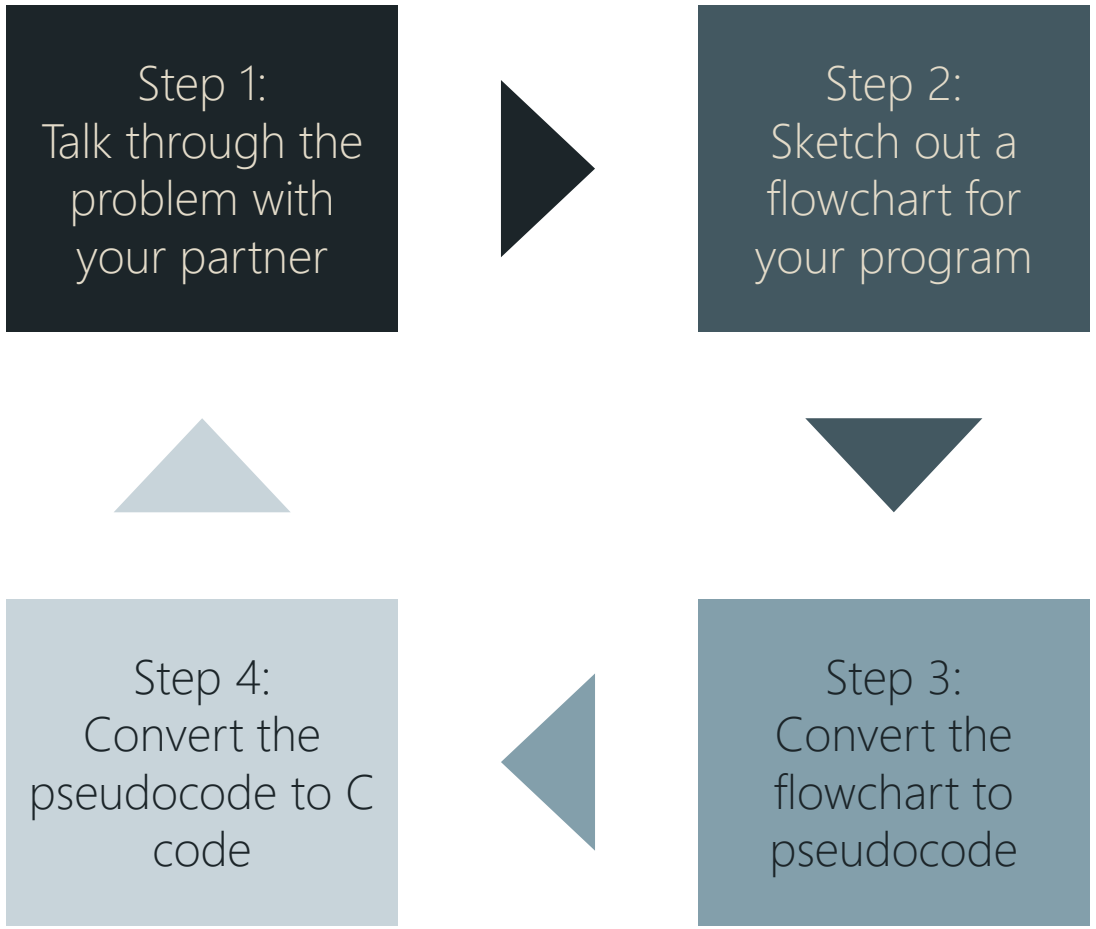
Rules of leap years:

1. Years divisible by 4 are leap years (e.g. 1904)
2. Except, years divisible by 100 are **NOT** leap years (e.g. 1900)
3. Except, years divisible by 400 are **ALWAYS** leap years (e.g. 2000)

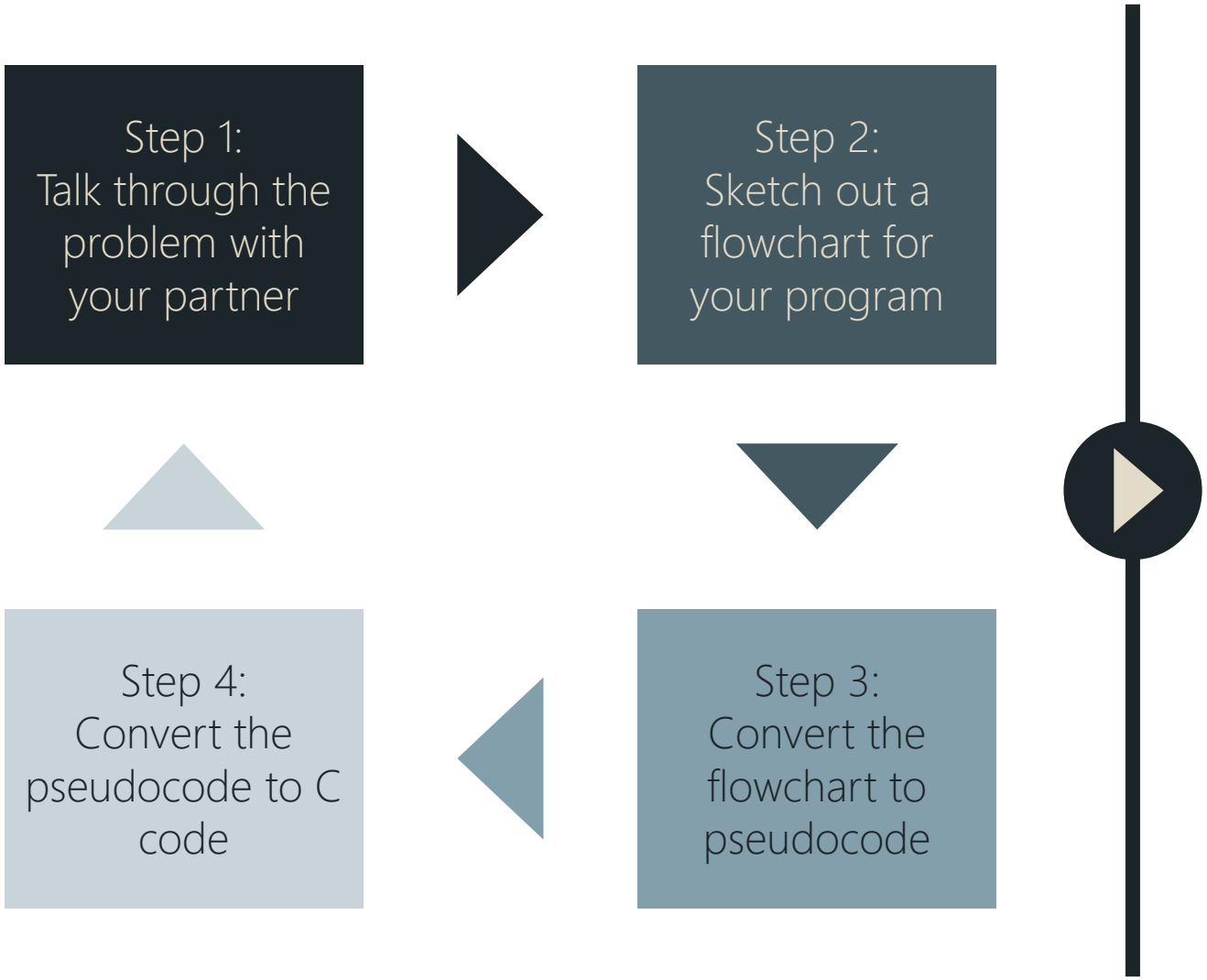
...this is how I would do it.



In pairs, attempt the programming exercise



In pairs, attempt the programming exercise



Problem:

1. Prompt the user for an integer
2. Scan in one integer **n**
3. If **n** is 5, print "Hurrah!"

In pairs, attempt the programming exercise

Problem:

1. Scan in two integers **a** and **b**
2. If the first integer is less than the second, print out a short error message **using a procedure**
3. If the second integer is 0, print out a different short error message
4. If the first integer is larger than the second, prints **`a / b`** and **`(a * 1.0) / (b * 1.0)`**