Pointers

- Pointers are memory addresses to variables / values.
- All variables have a pointer, which is the “hidden” memory address (just a number) where the computer is storing it in memory.
- For now, just consider memory to be a long list of “boxes”, each box having its own address and space for a value (where variables are stored)

Pointer Syntax and Basics:

```
// declare n an integer
// set value of variable named "n" to 15
int n = 15;
// let's pretend that the computer decides to store "n"
// in address 523

// int *p means declare "p" to be pointer to an int
// &n means get the memory address to variable "n"
int *p = &n; // p = 523

// when we use * before a variable after it has been declared
// it is treated as a value that we can set
int z = *p; // z = 15

// when * used on the left hand side of equals sign
// we change/set the value of the variable at that pointer
*p = 7; // n = 7, p = 523
// notice how p doesn't change, but n and *p change
```

We can use pointers to change the value of variables outside of functions.

Here’s an example:

```
int add_by_value(int a) {
  a = a + 1;
  return a;
}

int add_by_ref(int *b) {
  *b = *b + 1;
  return *b;
}
```

In main:

```
int x = 5;
int m = add_by_value(x);
```

```
print(m); // 6
print(x); // 5

// notice how x is unchanged when passed by value
// this means that the value is "copied" into the function

int y = 7;
int n = add_by_ref(&y); // pass in y's memory address / pointer

print(n); // 8
print(y); // 8

// notice how in this case, y is changed
// this is because we passed in a pointer
// in the function, we edit the value at that pointer
```