Brief Recap: Tools we've learned so far:

- variables (int x;)
  gives space to write/remember something.
- assignment (x = y;)
  overwrite left side w/ whatever is on right side.
- basic "flow of control" (if ... + while...)
  More on that next time

Today: closer look at **datatypes**.

Remember: each variable must have a datatype,
(as does each expression, e.g. x+3)

Some basic datatypes:

- int ($-2^{31} \leq x < 2^{31}$) (4 bytes)
- long ($-2^{63} \leq x < 2^{63}$) (8 bytes on 64-bit cpu...)
- char ($-2^{7} \leq c < 2^{7}$) (1 byte)
- float (see ieee format... uses some form of scientific notation)
  4 bytes
- double (same as float but 8 bytes)

- size_t $(0 \leq x < 2^{64})$ ⊛ # bytes depends on size of CPU registers.

```
size_t x = 2^64 - 1;

    x = x + 2;
    cout << x; // prints 1
```

Properties of Numeric types (int, float, etc..)

"Closure": $\text{int} \; \boxed{?} \; \text{int} \longrightarrow \text{int}$

$\nearrow$

+, -, *, / ...

```
cout << 6/7; // prints 0
```

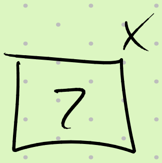
"Contamination":

$\text{int} \; \boxed{?} \; \text{float} \longrightarrow \text{float}$

Warning: watch out for "rounding error" ...

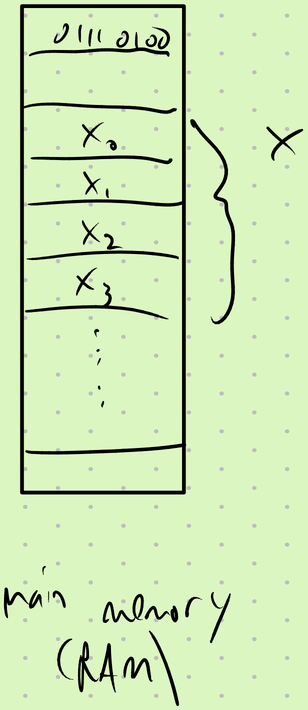
A few alternate pictures:

```
int x;
x = 7;
```



unsigned int x;

// $x = x_0 + x_1 \cdot 256 + x_2 \cdot 256^2 + x_3 \cdot 256^3$



<u>Note</u> for integer types, there are <u>unsigned</u> variants!

E.g., unigned int x; // $0 \leq x < 2^{32}$

unsigned long y; // $0 \leq y < 2^{64}$