C* Integers and Pointers

- C* integers are unsigned 64-bit integers
- In C* there are five arithmetic operators: `+`, `-`, `*`, `/`, `%`
- And six comparison operators: `==`, `!=`, `<`, `<=`, `>`, `>=`
- C* pointers are 64-bit pointers to C* integers
- And pointer arithmetic: `+`, `-`

C* versus C Integer Literals

- C* integer literals are unsigned 64-bit
- C integer literals are signed 32-bit
- For example, `1/-1==0` in C* but `1/-1==-1` in C
- And, `1%-1==1` in C* but `1%-1==0` in C
- Also, `1<-1` and `1<=-1` hold in C* but `1>-1` and `1>=-1` do not whereas the opposite is true in C
- The semantics of `/` and `%` as well as <, <=, >, and >= is different for signed and unsigned integers!

C* Integers and Pointers

- C* integers are unsigned 64-bit integers
- In C* there are five arithmetic operators: `+`, `-`, `*`, `/`, `%`
- And six comparison operators: `==`, `!=`, `<`, `<=`, `>`, `>=`
- C* pointers are 64-bit pointers to C* integers
- And pointer arithmetic: `+`, `-`