C* by Example

```
uint64_t atoi(uint64_t* s) {
  uint64_t n;

  n = 0;

  // loop until s is terminated
  while (*s != 0) {
    // use base 10, offset by '0'
    n = n * 10 + *s - '0';

    // go to next digit
    s = s + 1;
  }

  return n;
}
```

procedure

return type

formal parameter

local variable

assignment

return statement

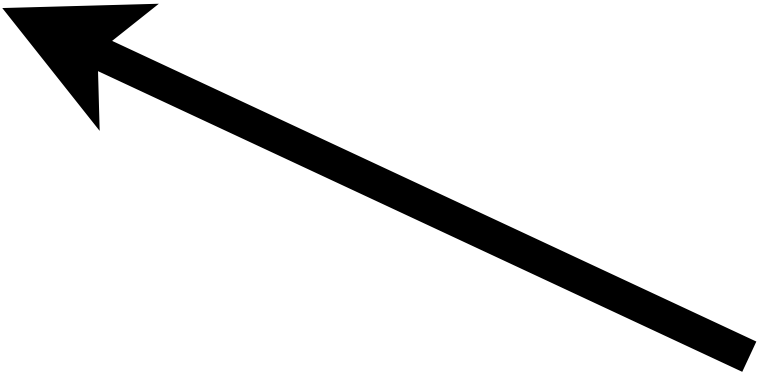
expression

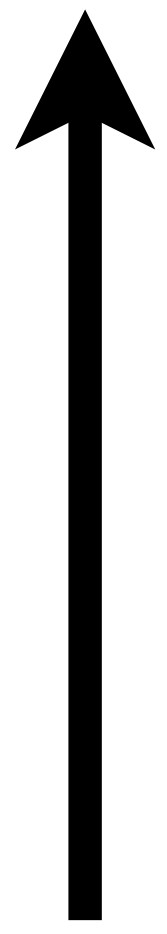
pointer arithmetic

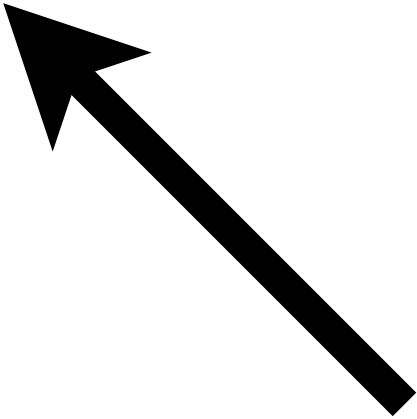
integer arithmetic

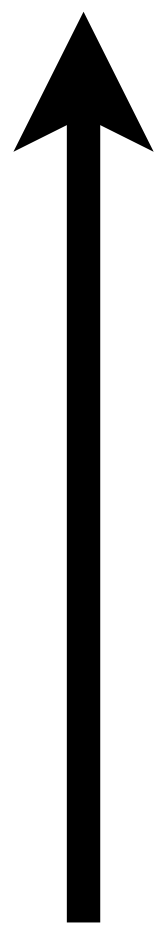
integer literal

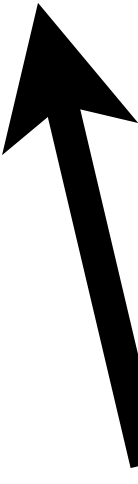
character literal

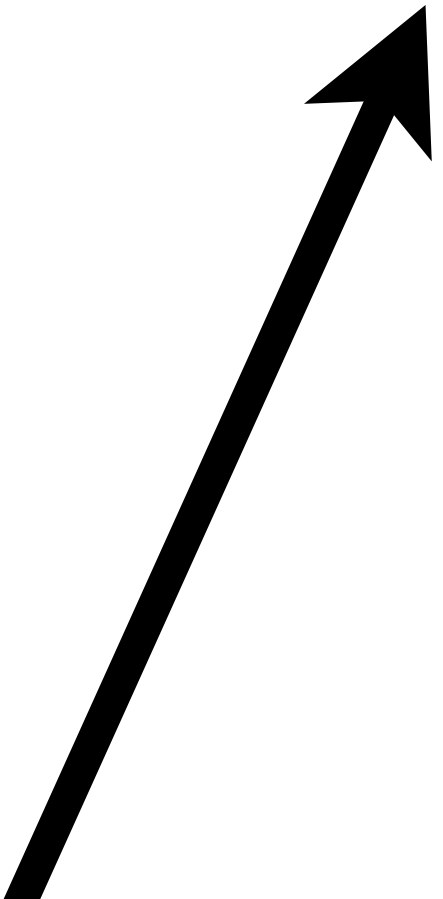


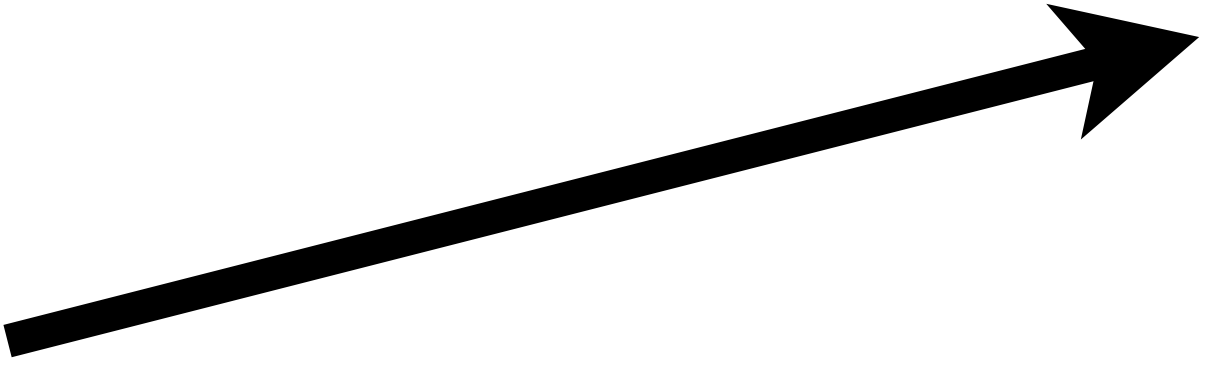




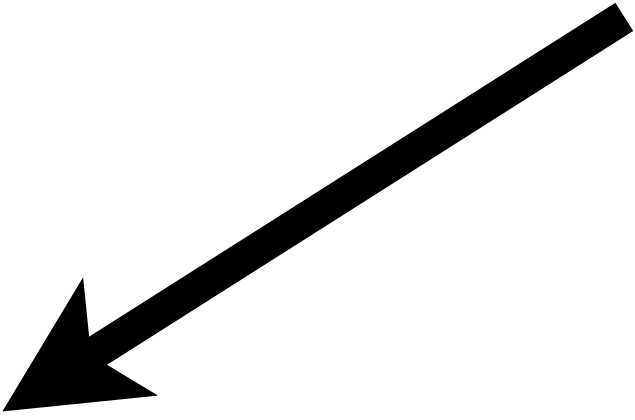


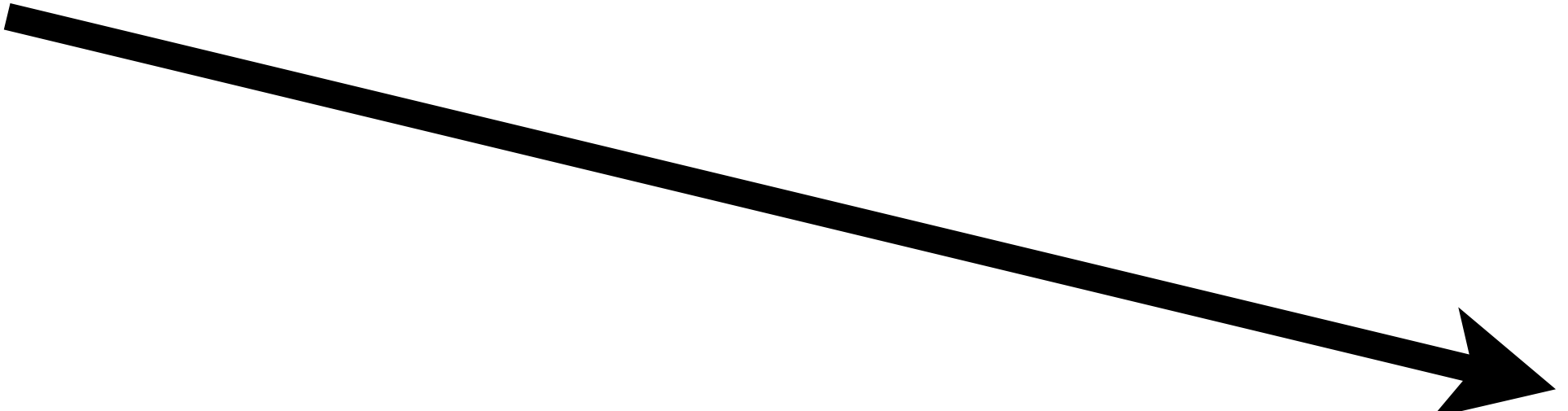


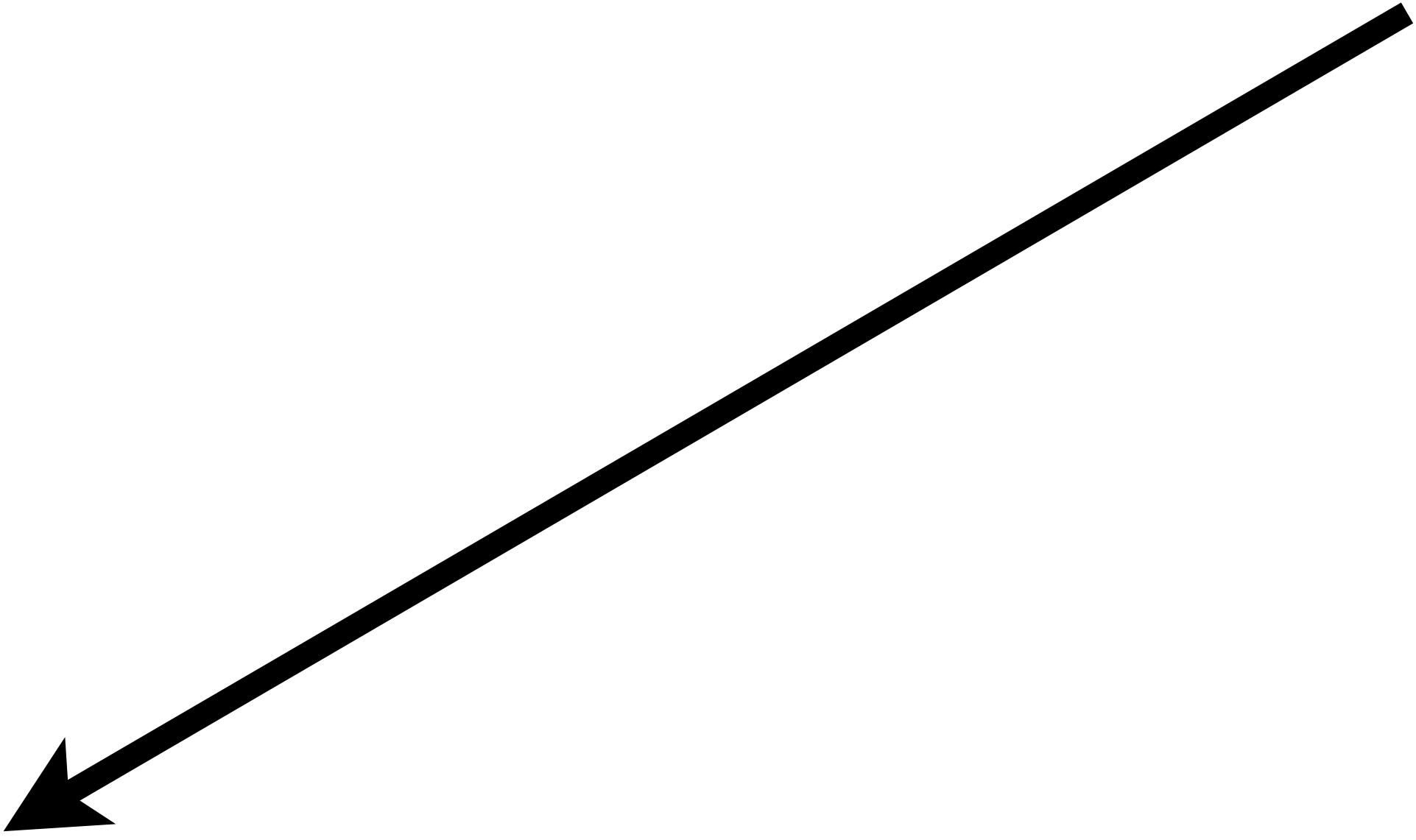




dereferencing

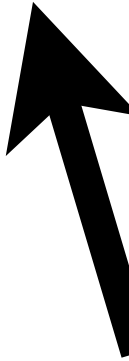






while loop

pointer type



integer type

comparison

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* by Example

```
uint64_t atoi(uint64_t* s) {
  uint64_t n;

  n = 0;

  // loop until s is terminated
  while (*s != 0) {
    // use base 10, offset by '0'
    n = n * 10 + *s - '0';

    // go to next digit
    s = s + 1;
  }

  return n;
}
```

procedure

pointer type

formal parameter

return type

integer type

local variable

integer literal

dereferencing

assignment

comparison

while loop

expression

integer arithmetic

character literal

pointer arithmetic

return statement