Operator ++ for Pointers
for Pointers

• Same idea...
++ for Pointers

• Same idea...
• ...but: value of pointer is an **address**.
• Same idea...
• ...but: value of pointer is an **address**.
  → Shift pointer to **next object**.
++ptr
++ptr
Exercise – Applying Pointers
Exercise – Applying Pointers

• Apply this function...

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint
// valid ranges
void f (int* b, int* e, int* o) {
  while (b != e) {
    --e;
    *o = *e;
    ++o;
  }
}
```

• ...to this example-array:

```
 1 3 -8 1 5 -3 4
```
Exercise – Applying Pointers

```c
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
Exercise – Applying Pointers

```c
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
Exercise – Applying Pointers

```c
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}

(From: Script Exercise 113)
Exercise – Applying Pointers

void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}

(From: Script Exercise 113)
Exercise – Applying Pointers

void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}

(From: Script Exercise 113)
Exercise – Applying Pointers

```c
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}`
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}

(From: Script Exercise 113)
Exercise – Applying Pointers

```c
void f (int *b, int *e, int *o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
Exercise – Applying Pointers

• Now determine a POST-condition for the function.

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
Exercise – Applying Pointers

• Something like this:

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint
//       valid ranges
// POST: The range [b, e) is copied in reverse
//       order into the range [o, o+(e-b))
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
Exercise – Valid Inputs
Exercise – Valid Inputs

• Which of these inputs are valid?

```c
int a[5] = {1, 2, 3, 4, 5};

a) f(a, a+5, a+5);
b) f(a, a+2, a+3);
c) f(a, a+3, a+2);
```

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```
Exercise – Valid Inputs

• Which of these inputs are valid?

\[
\text{int } a[5] = \{1, 2, 3, 4, 5\};
\]

a) \( f(a, a+5, a+5); \) \( \times \)

b) \( f(a, a+2, a+3); \)

c) \( f(a, a+3, a+2); \)

\[
\text{void } f (\text{int* } b, \text{int* } e, \text{int* } o) \{
\text{while } (b != e) \{
\text{--e;}
\text{*o = *e;}
\text{++o;}
\}
\}
\]

// PRE: [b, e) and [o, o+(e-b)) are disjoint
// valid ranges
Exercise – Valid Inputs

• Which of these inputs are valid?

```c
int a[5] = {1, 2, 3, 4, 5};
a) f(a, a+5, a+5);  X
b) f(a, a+2, a+3);  ✓
c) f(a, a+3, a+2);
```

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint
// valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

[0, o+(e-b)] is out of bounds
Exercise – Valid Inputs

• Which of these inputs are valid?

int a[5] = {1, 2, 3, 4, 5};
a) f(a, a+5, a+5); ✗
b) f(a, a+2, a+3); ✓
c) f(a, a+3, a+2); ✗

// PRE: [b, e) and [o, o+(e-b)) are disjoint
// valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}

[0, o+(e-b)) is out of bounds
Ranges not disjoint

(From: Script Exercise 113)
Exercise – \texttt{const} Correctness
Exercise – const Correctness

• Make the function const-correct.

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

(From: Script Exercise 113)
Exercise – const Correctness

• Make the function const-correct.

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint valid ranges
void f (const int* const b, const int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

const: no write-access to target const: no shifts of pointer
By the way...
By the way...

• ...that’s the same function:

```c
// PRE: [b, e) and [o, o+(e-b)) are disjoint
//      valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) *(o++) = *(--e);
}
```