Operator ++ for Pointers

++ for Pointers

- Same idea...

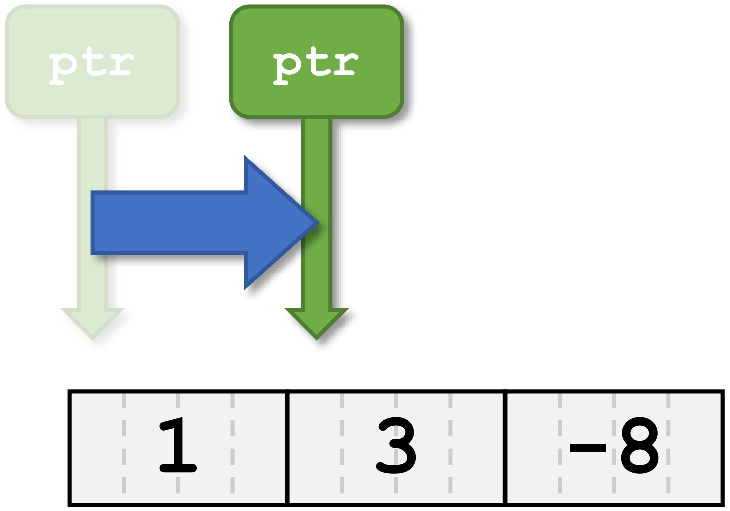
++ for Pointers

- Same idea...
- …but: value of pointer is an **address**.

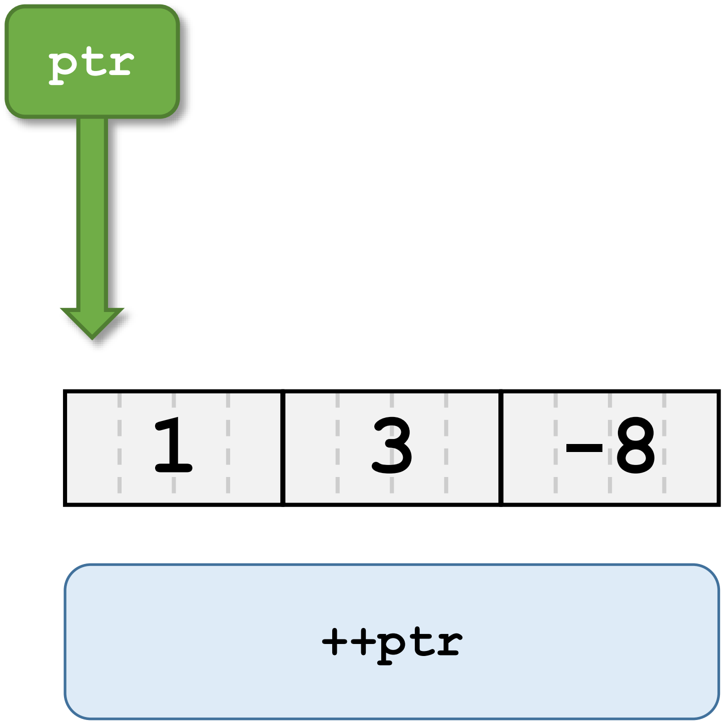
++ for Pointers

- Same idea...
- ...but: value of pointer is an **address**.

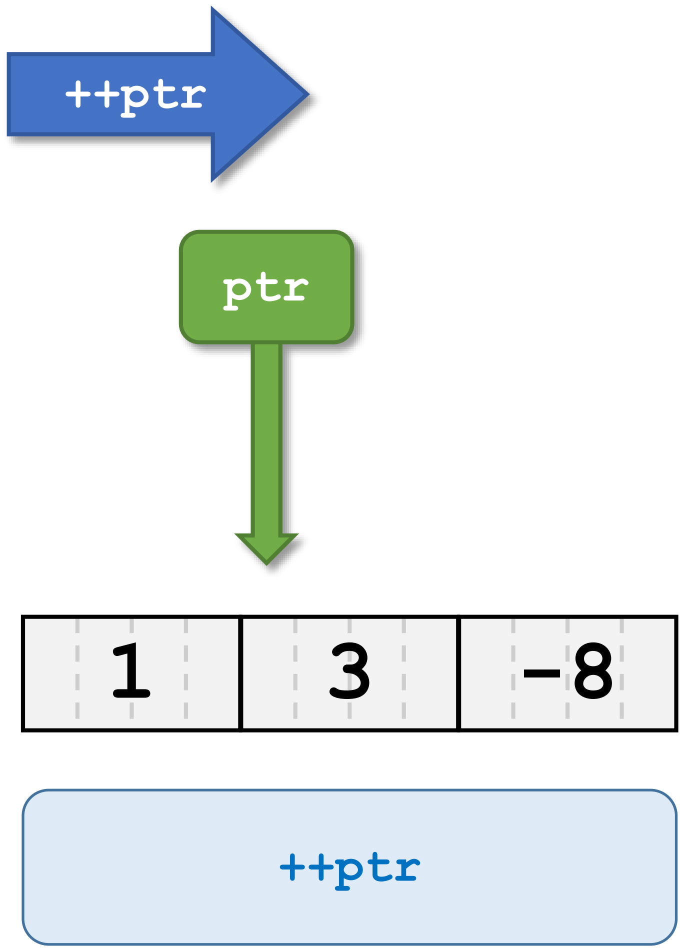
  → Shift pointer to **next object**.



++ptr



++ptr



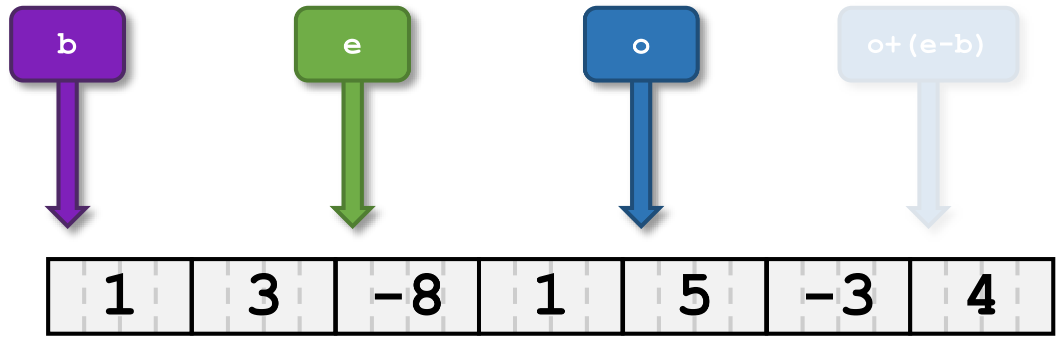
Exercise – Applying Pointers

Exercise – Applying Pointers

- Apply this function…

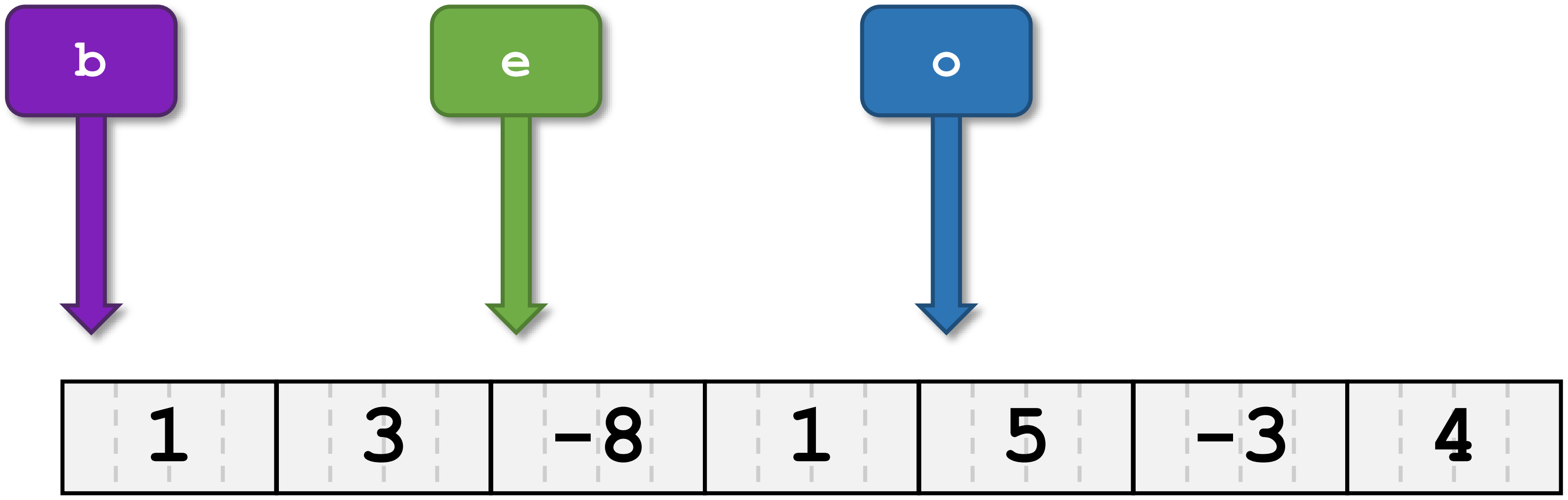
```
// PRE: [b, e) and [o, o+(e-b)) are disjoint
//      valid ranges
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```

- … to this example-array:



Exercise – Applying Pointers

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

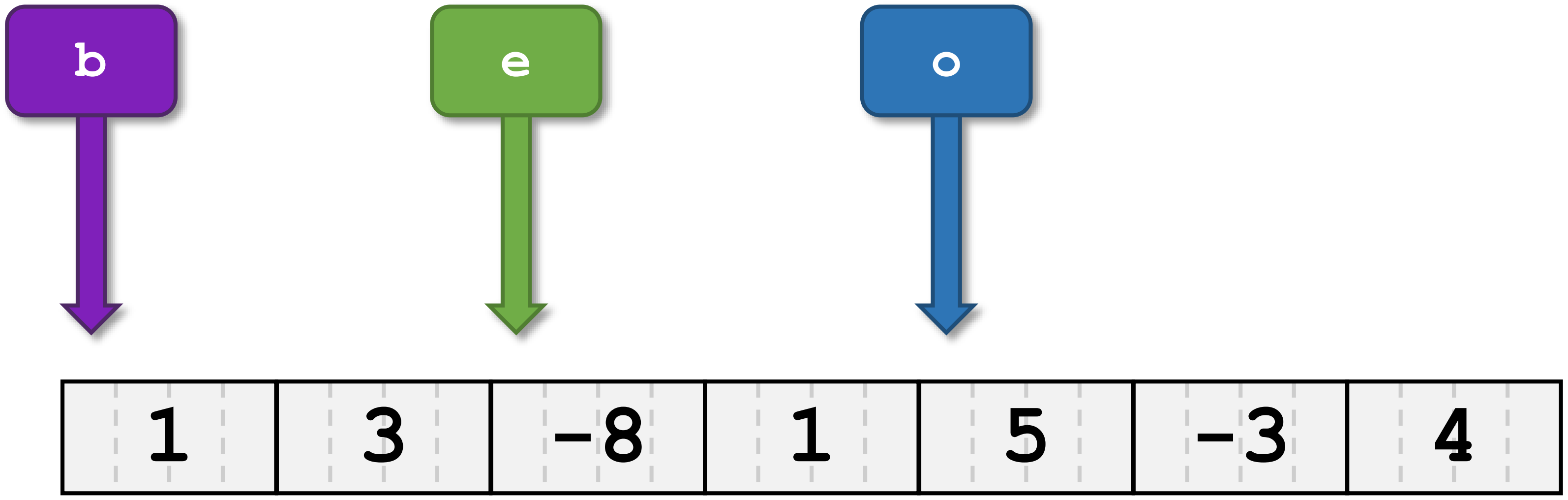


(From: Script Exercise 113)

Exercise – Applying Pointers

true

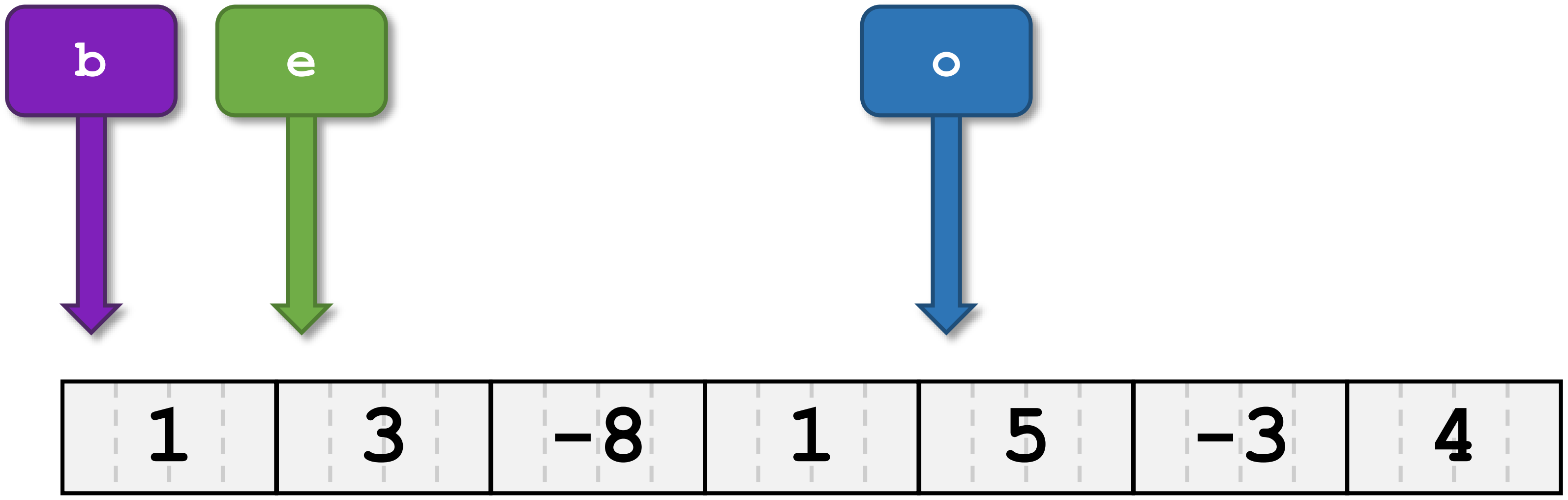
```
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```



(From: Script Exercise 113)

Exercise – Applying Pointers

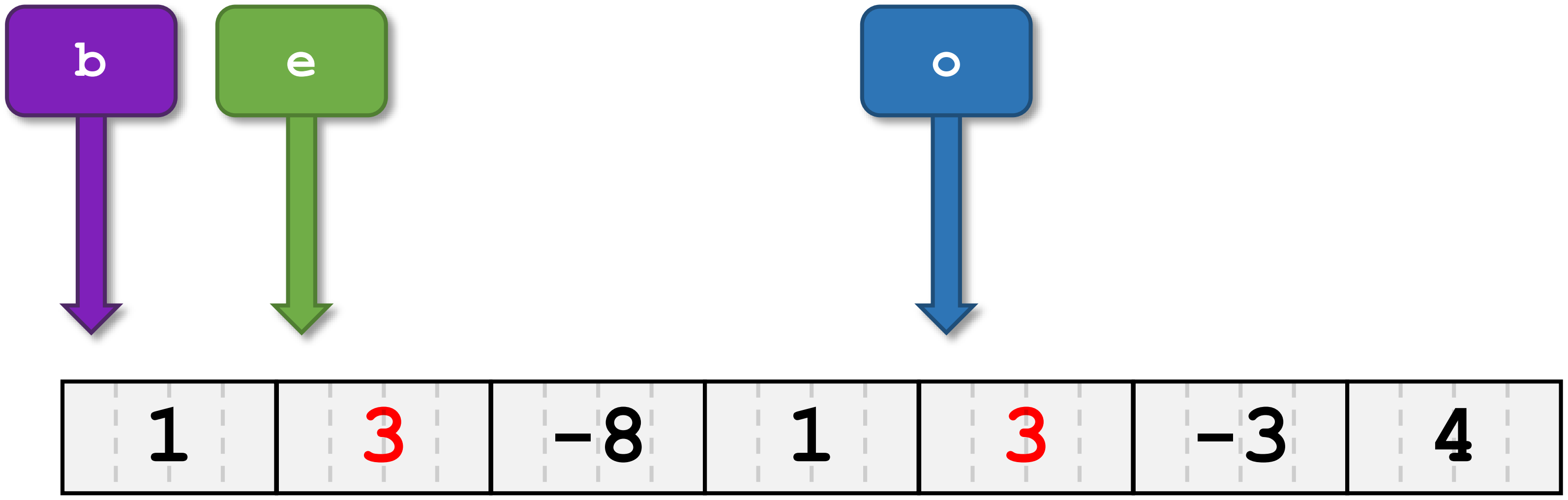
```
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```



(From: Script Exercise 113)

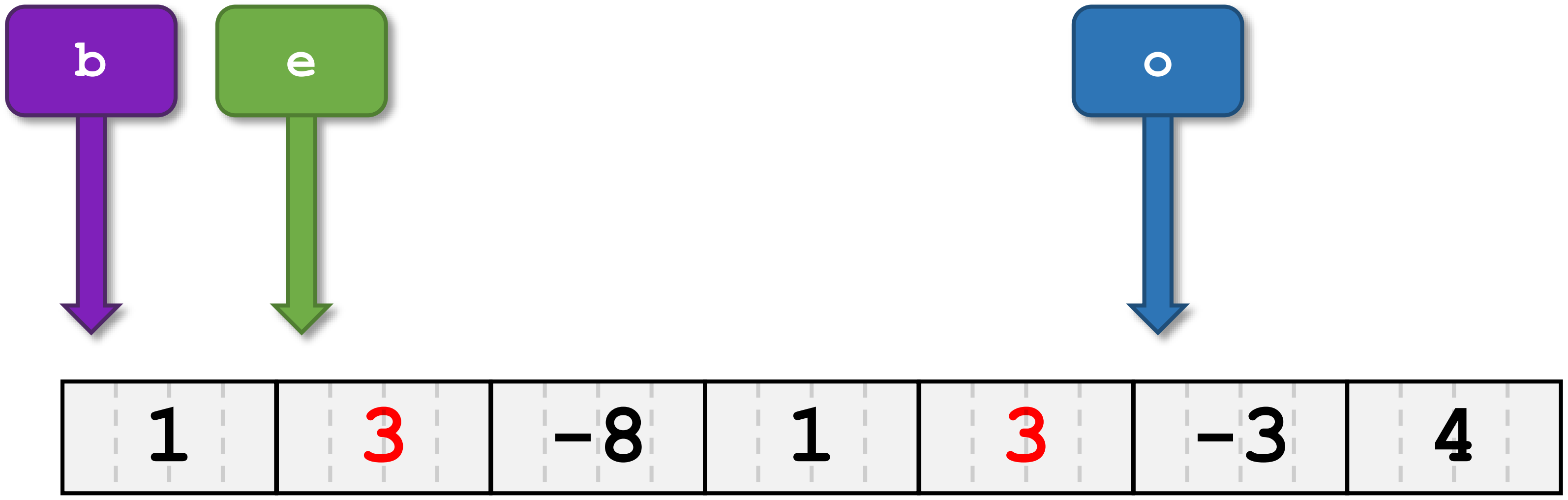
Exercise – Applying Pointers

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



Exercise – Applying Pointers

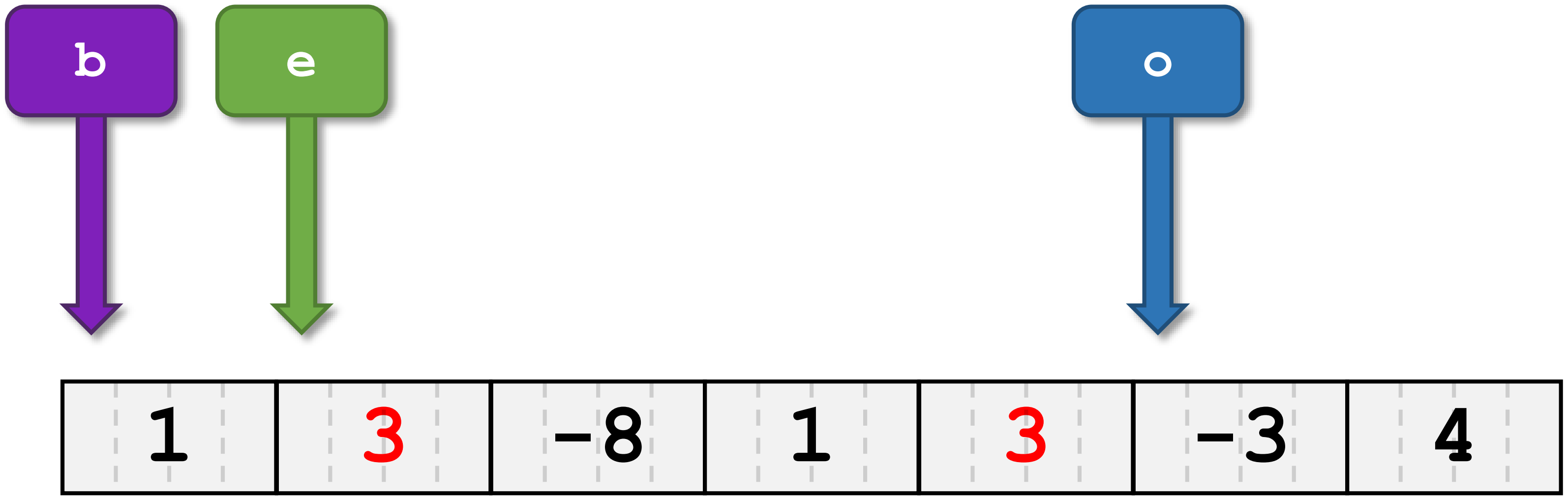
```
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```



Exercise – Applying Pointers

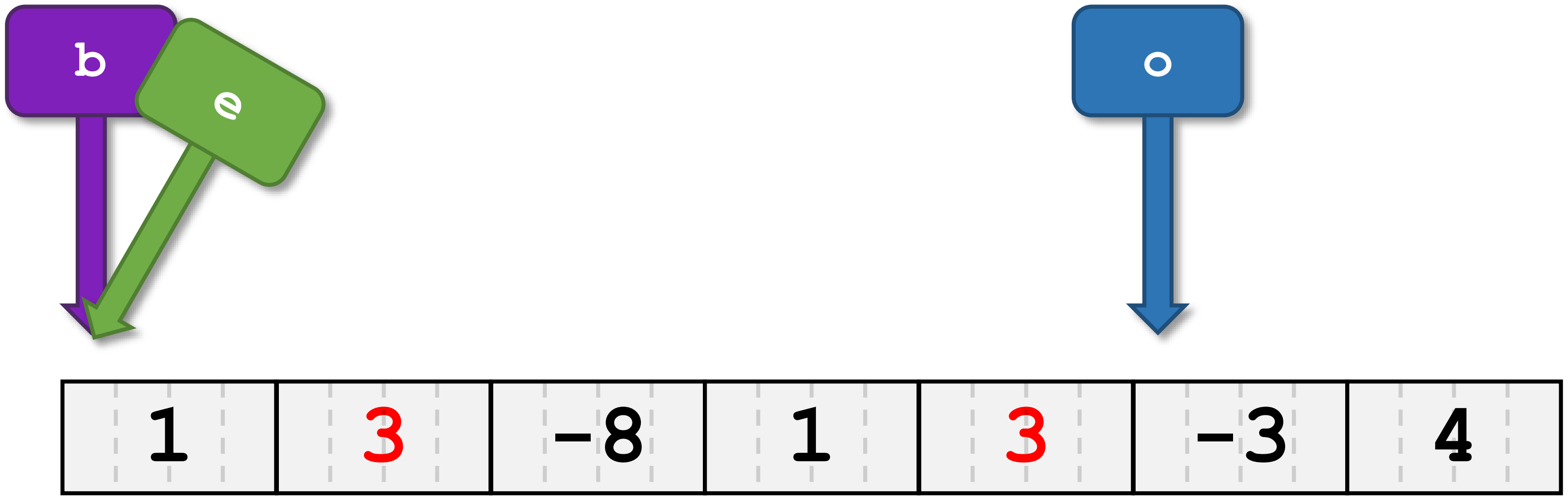
true

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



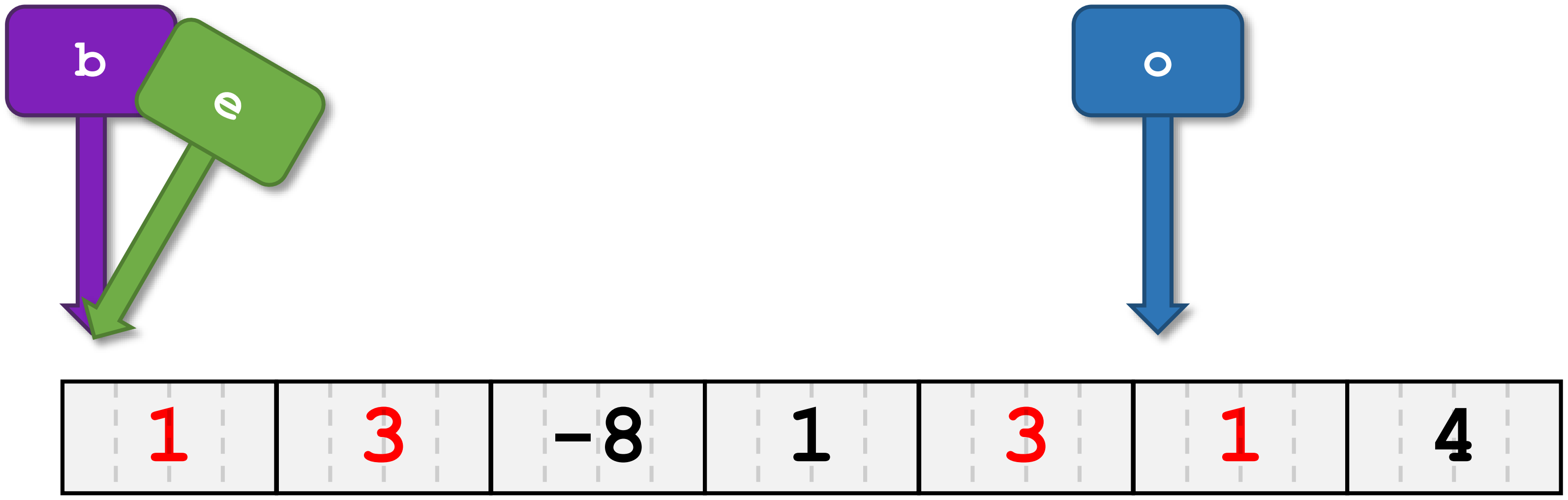
Exercise – Applying Pointers

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



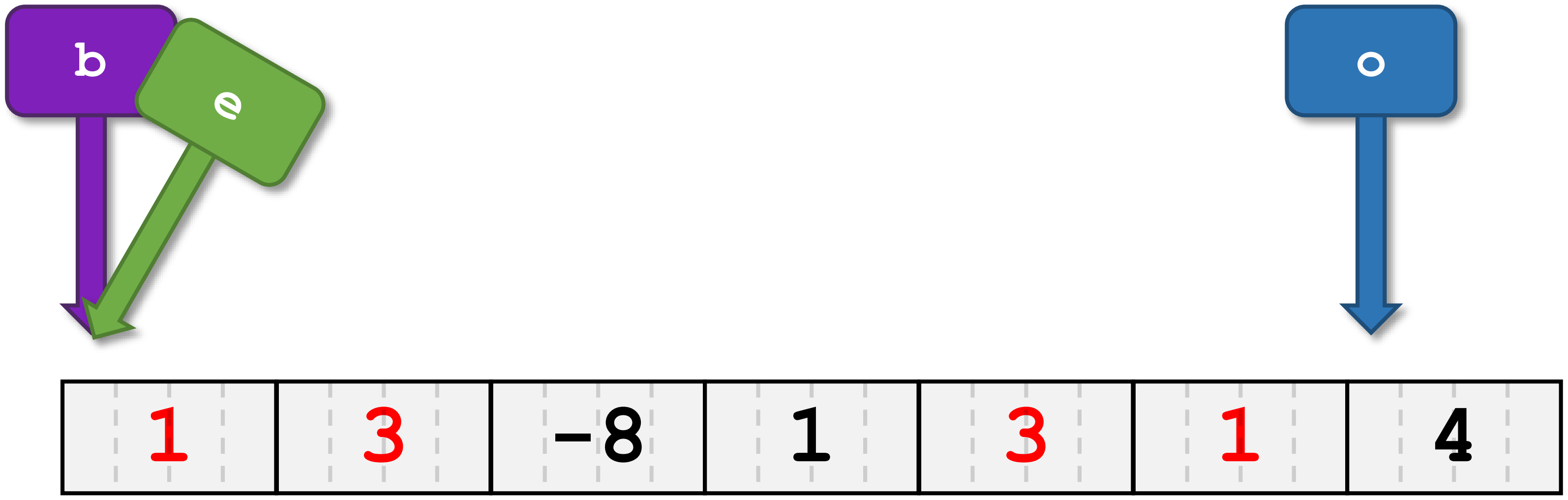
Exercise – Applying Pointers

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



Exercise – Applying Pointers

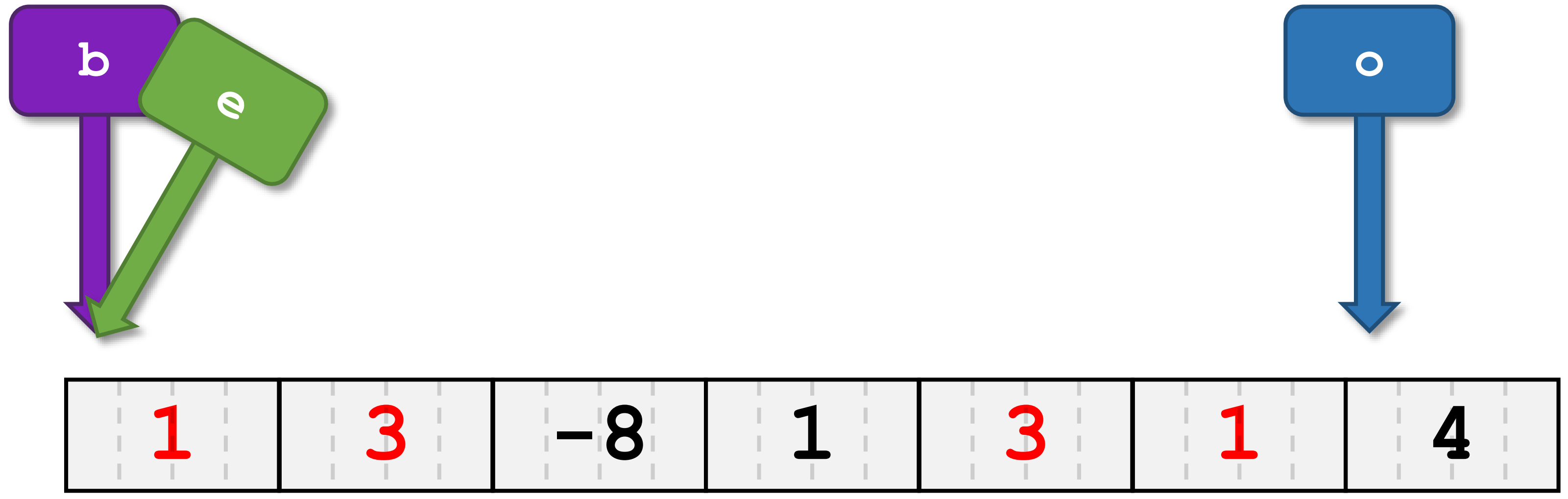
```
void f (int* b, int* e, int* o) {
    while (b != e) {
        --e;
        *o = *e;
        ++o;
    }
}
```



Exercise – Applying Pointers

false

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



Exercise – Applying Pointers

- Now determine a POST-condition for the function.

```
// 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 – Applying Pointers

- Something like this:

```
// 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;
    }
}
```

Exercise – Valid Inputs

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;
    }
}
```

(From: Script Exercise 113)

Exercise – Valid Inputs

- Which of these inputs are valid?

`[o,o+(e-b))` is **out of bounds**

```
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);
```

```
// 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 – 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);
```

`[o,o+(e-b))` is **out of bounds**

```
// 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 – 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);    ✗
```

`[o,o+(e-b))` is **out of bounds**

Ranges **not disjoint**

```
// 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 – `const` Correctness

Exercise – `const` Correctness

- Make the function `const`-correct.

```
// 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.

`const`: no write-access to **target**
`const`: no shifts of **pointer**

```
// 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;
    }
}
```

(From: Script Exercise 113)

By the way…

By the way…

- …that's the same function:

```
// 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);
}
```