

Pointer Exercise

Pointer Exercise

- Determine a POST-condition for 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) *(o++) = *(--e);
}
```

Pointer Exercise

- Determine a POST-condition for this function.

```
// 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) *(o++) = *(--e);
}
```

Pointer Exercise

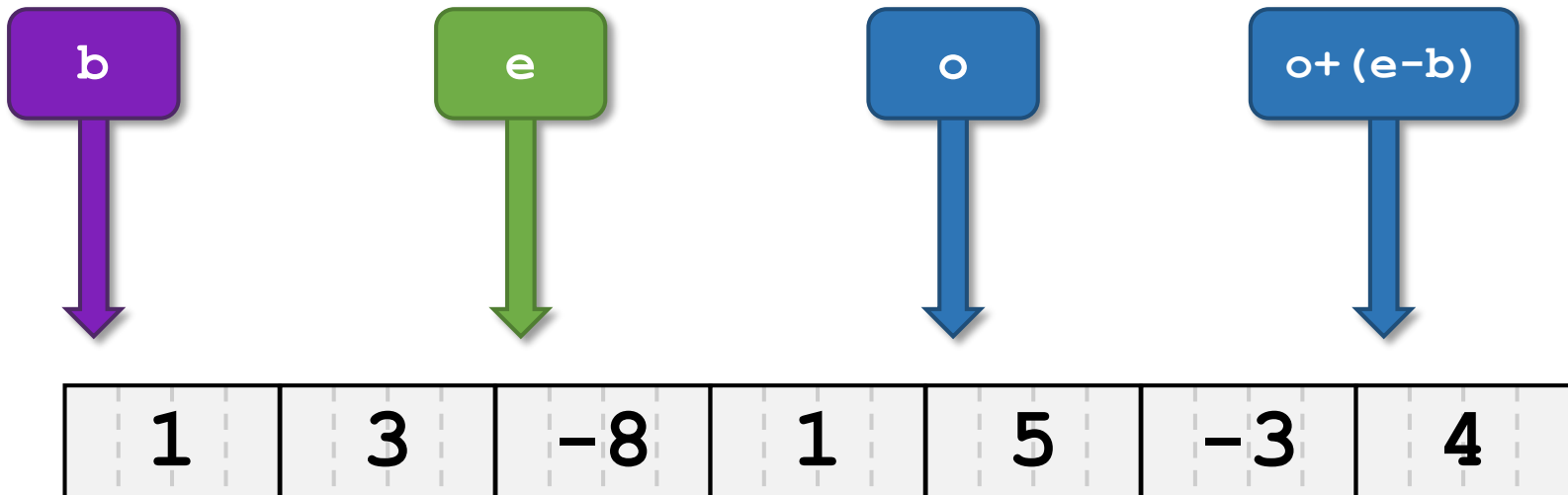
- Determine a POST-condition for this function

```
// PRE: [b, e) and [o, o+(e-b))  
//       disjoint valid ranges  
// POST: The range [b, o) is the  
//        reverse of the  
//        range [o, o+(e-b))  
void f (int b, int e, int o, int t)  
{  
    while (b < o) *o++ = *b++;  
    while (o < t) *o++ = *(--e);  
}
```

But why?

Pointer Exercise

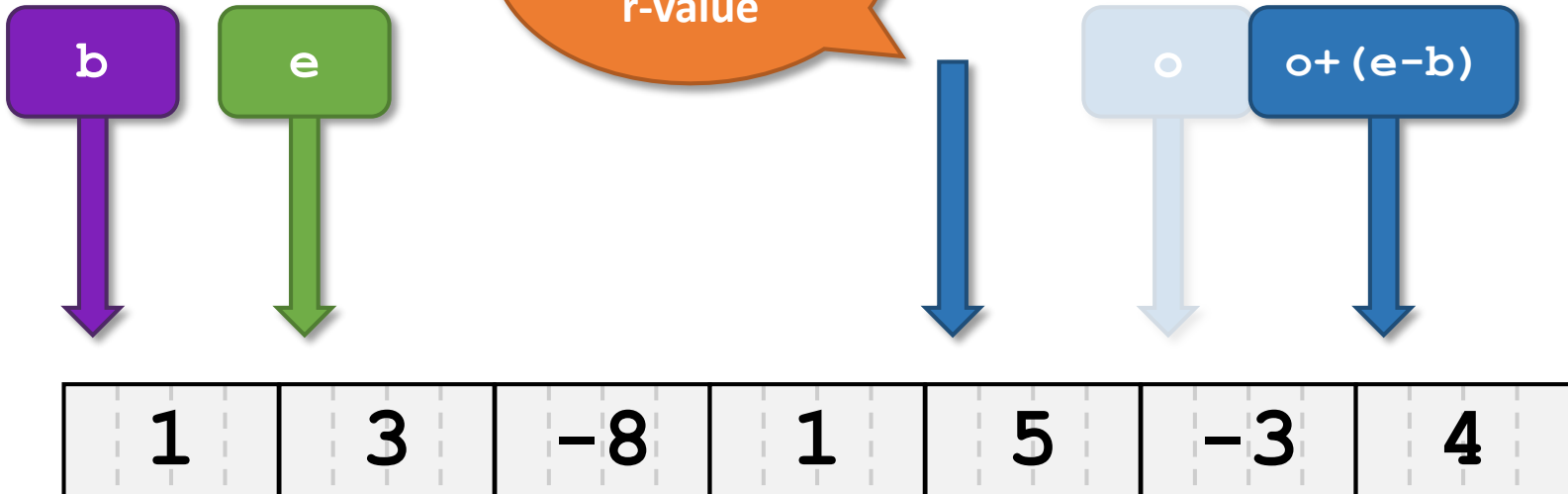
```
void f (int* b, int* e, int* o) {  
    while (b != e) *(o++) = *(--e);  
}
```



Pointer Exercise

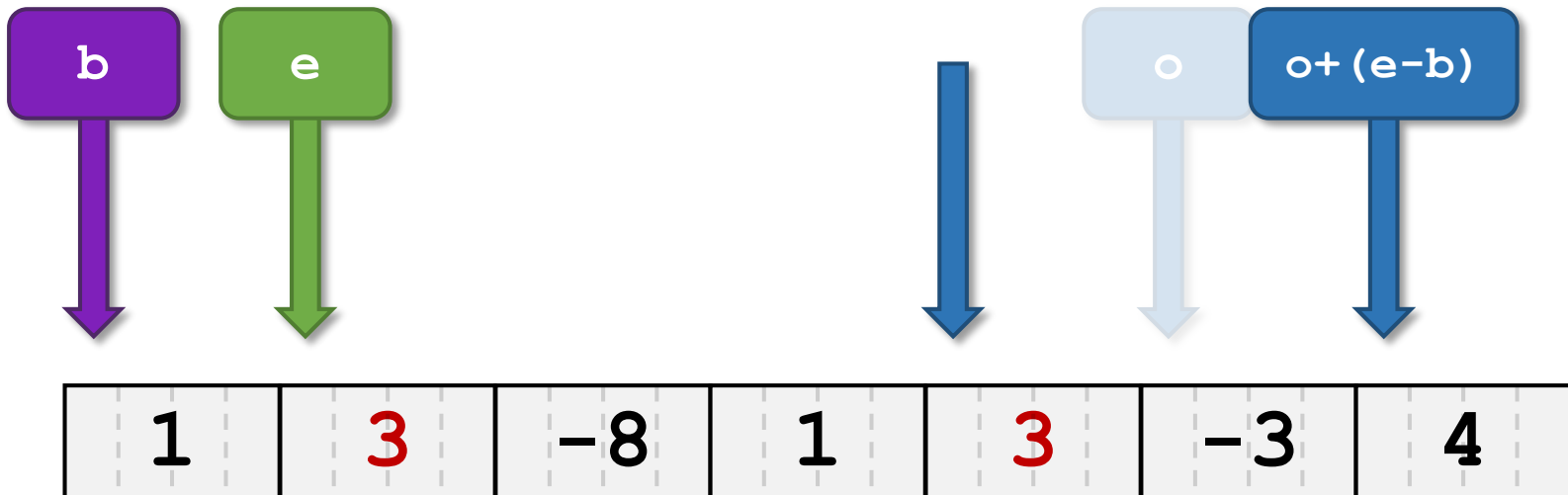
```
void f (int* b, int* e, int* o) {  
    while (b != e) *(o++) = *(--e);  
}
```

`o++` returns
pointer-
r-value



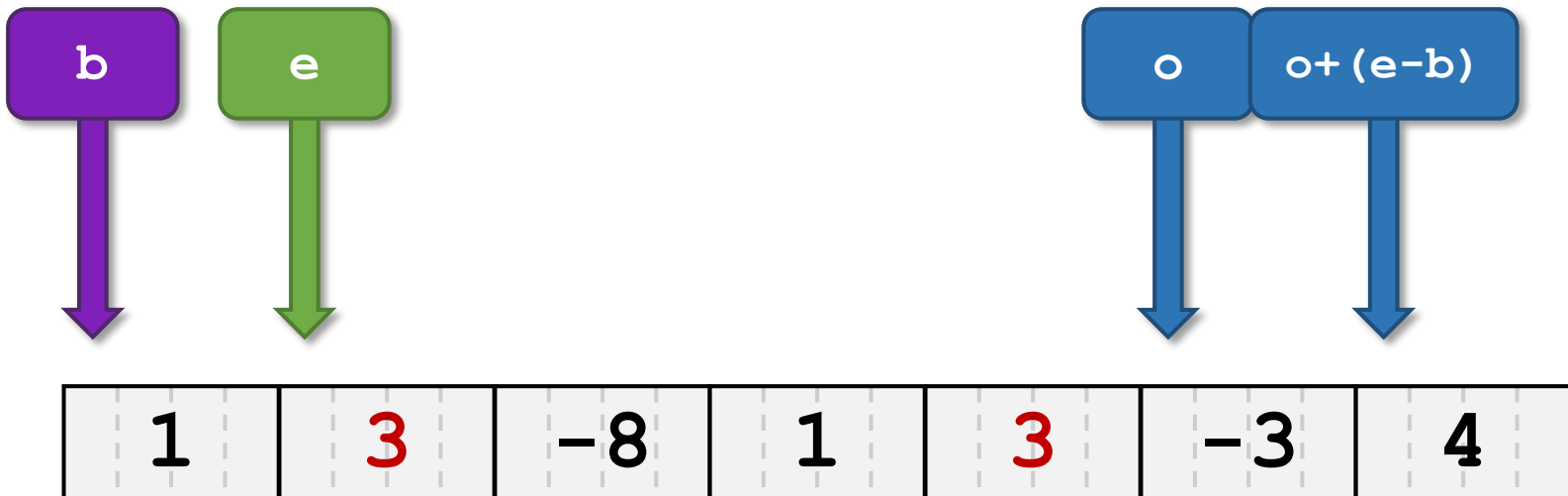
Pointer Exercise

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



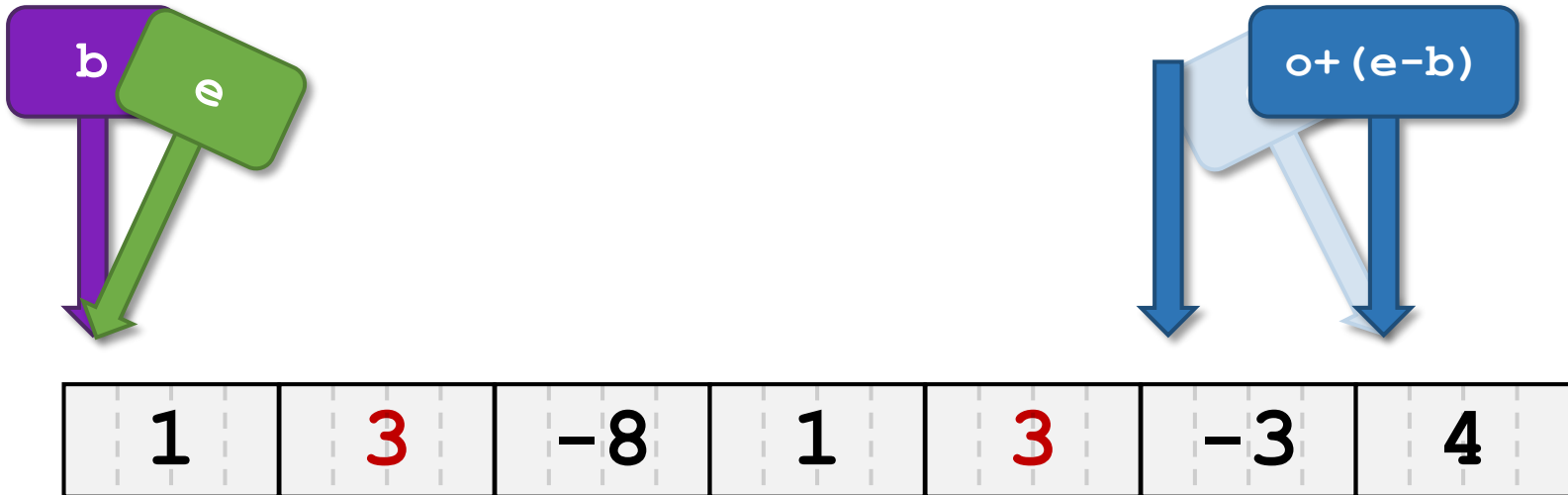
Pointer Exercise

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



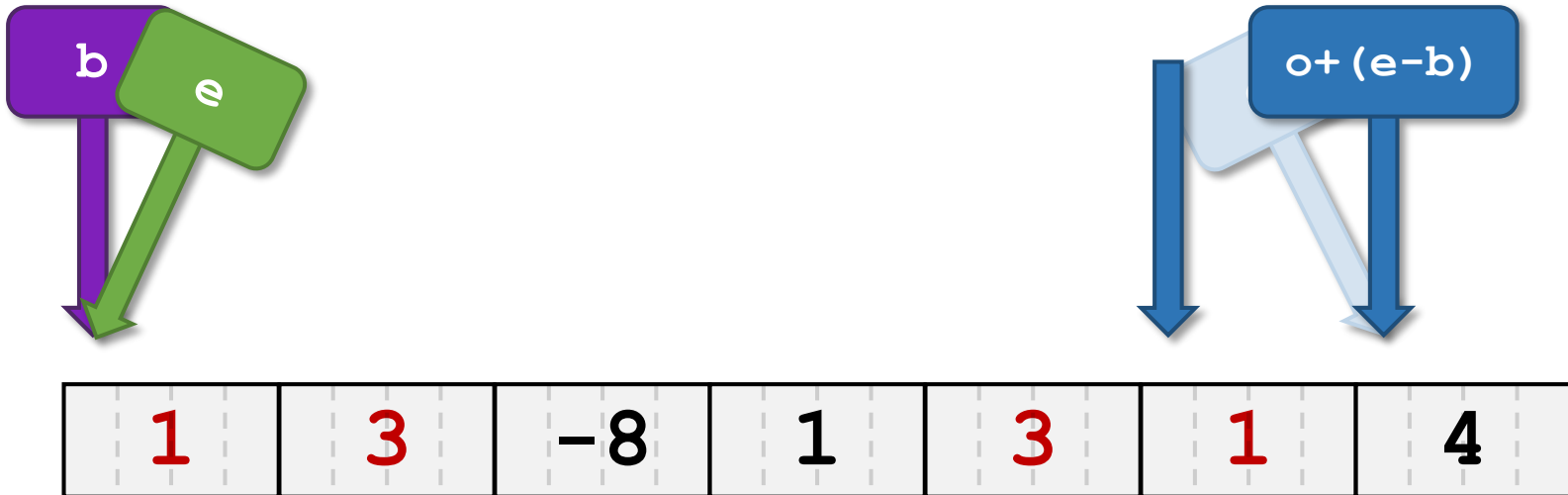
Pointer Exercise

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



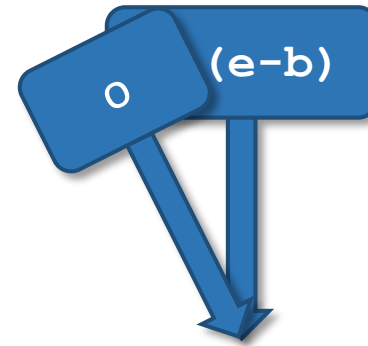
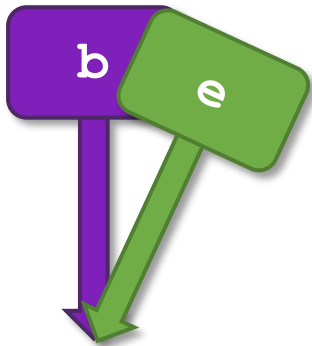
Pointer Exercise

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



Pointer Exercise

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



1	3	-8	1	3	1	4
---	---	----	---	---	---	---

Valid Inputs

Pointer Exercise

- 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) *(o++) = *(--e);  
}
```

Pointer Exercise

- Which of these inputs are valid?

```
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);
```

$[o, o+(e-b))$
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) *(o++) = *(--e);  
}
```

Pointer Exercise

- Which of these inputs are valid?

```
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);
```

$[o, o+(e-b))$
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) *(o++) = *(--e);  
}
```

Pointer Exercise

- Which of these inputs are valid?

```
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); X
```

$[o, o+(e-b))$
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) *(o++) = *(--e);  
}
```


const **Correctness**

Pointer Exercise

- 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) *(o++) = *(--e);
}
```

Pointer Exercise

- Make the function `const`-correct.

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

Pointer Exercise

- Remark: the other `const` would be:

`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) *(o++) = *(--e);  
}
```