

# Pointers on Arrays

# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};
int* ptr = a;           // array-to-pointer conv
++ptr;                 // shift to the right
int my_int = *ptr;     // read target
ptr += 2;              // shift by 2 elements
*ptr = 18;             // overwrite target
int* past = a+5;
std::cout << (ptr < past) << "\n"; // compare pointers
```

-6	1	3	-8	1	5	-3	4	1	7	2
----	---	---	----	---	---	----	---	---	---	---

# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a;           // array-to-pointer conv  
++ptr;                 // shift to the right  
int my_int = *ptr;     // read target  
ptr += 2;              // shift by 2 elements  
*ptr = 18;             // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```



a

# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```

ptr



a

# Pointer Program

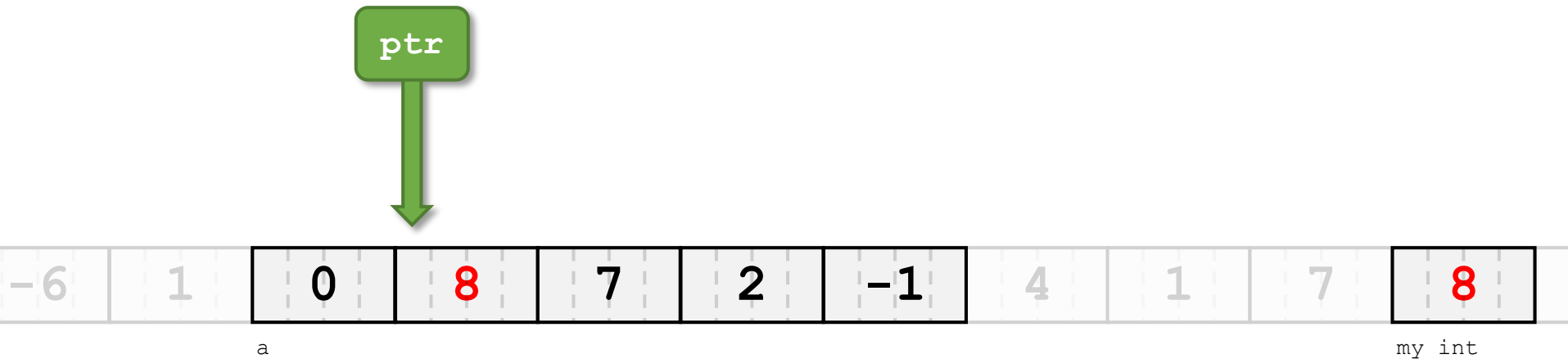
```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```



a

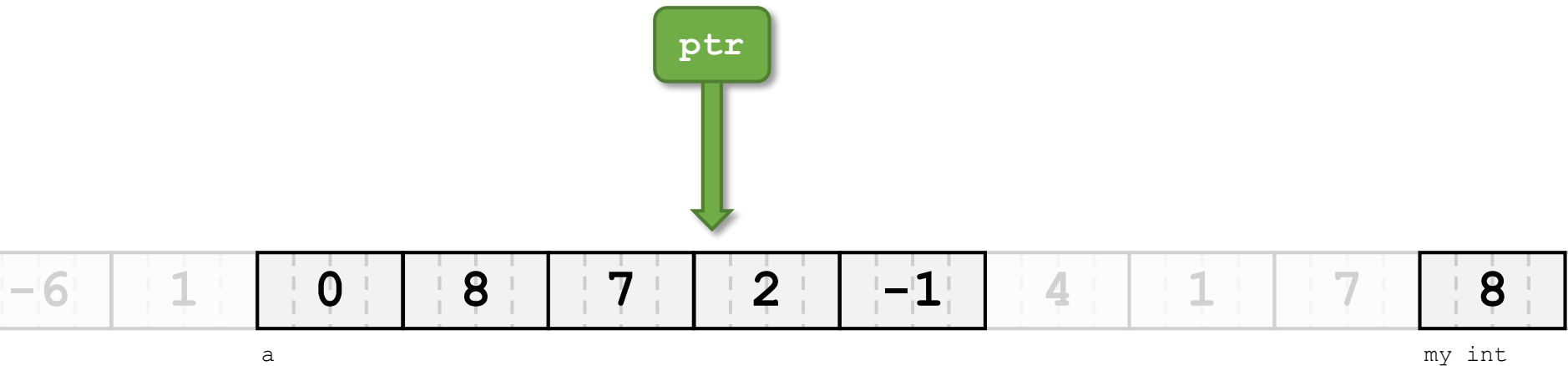
# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```



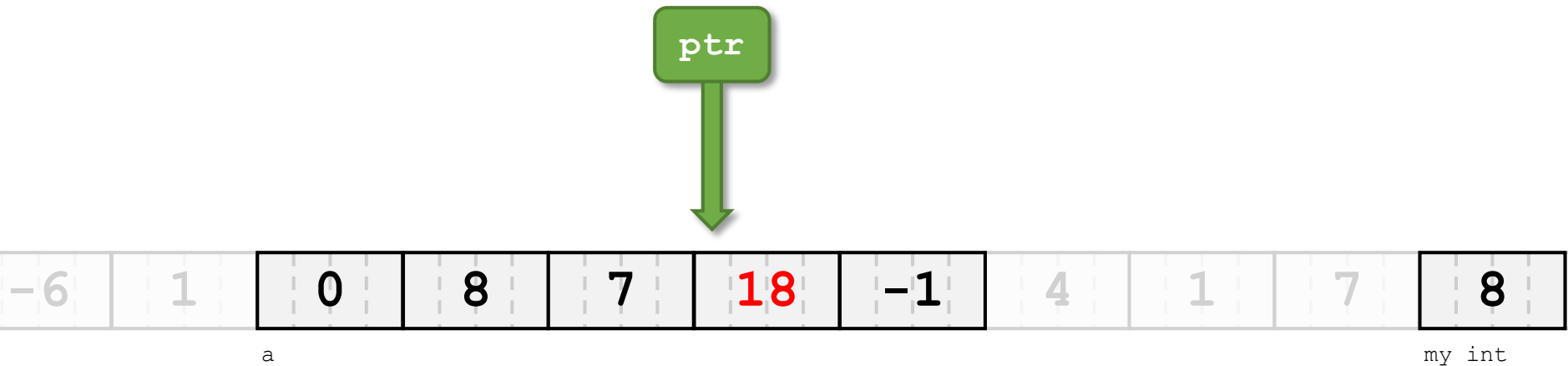
# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```



# Pointer Program

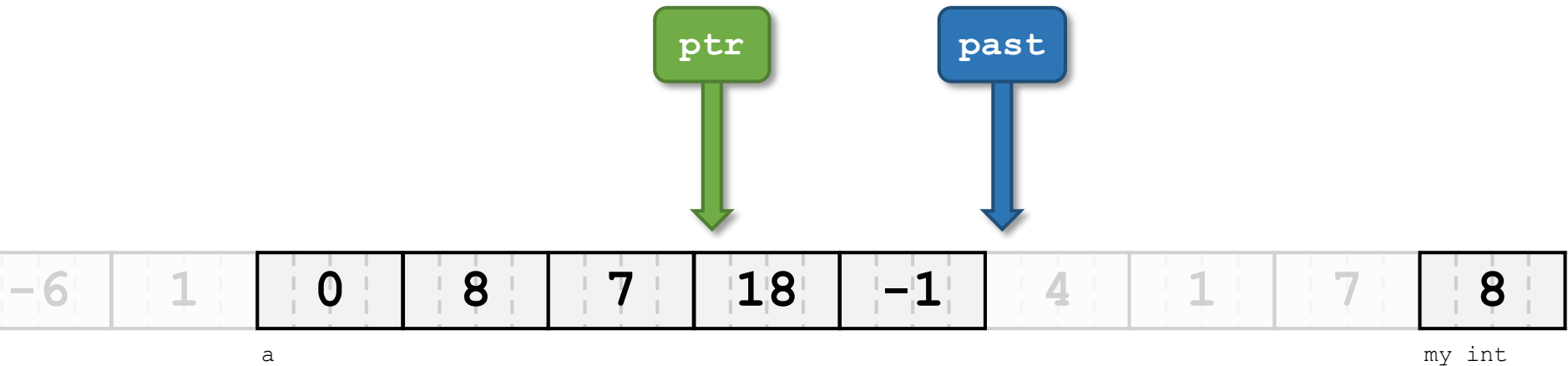
```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```





# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```

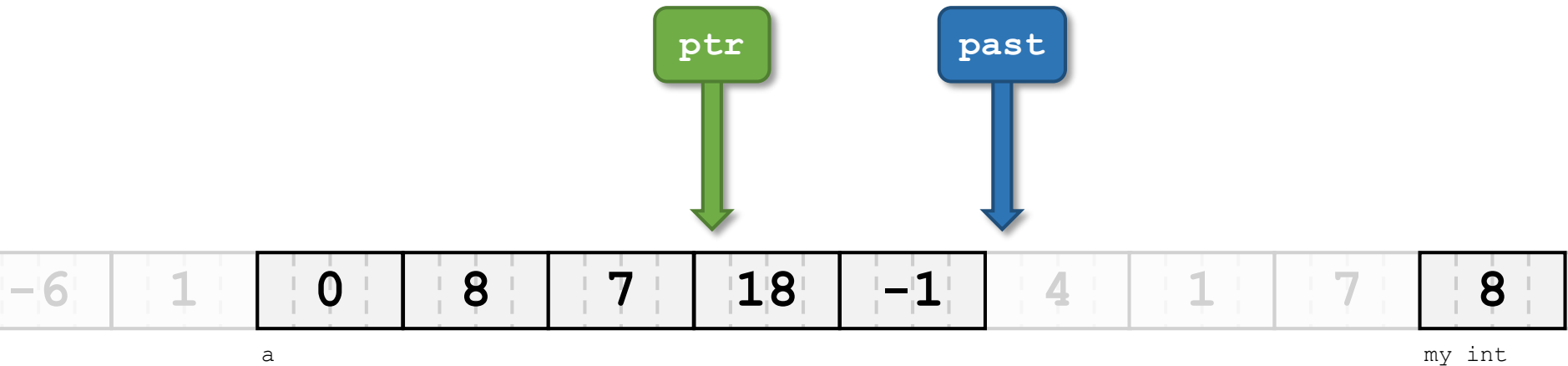


# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // arr  
++ptr; // shi  
int my_int = *ptr; // read  
ptr += 2; // shift  
*ptr = 18; // overw  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```

Output: true

Because ptr is  
"to the left" of past.



# Program

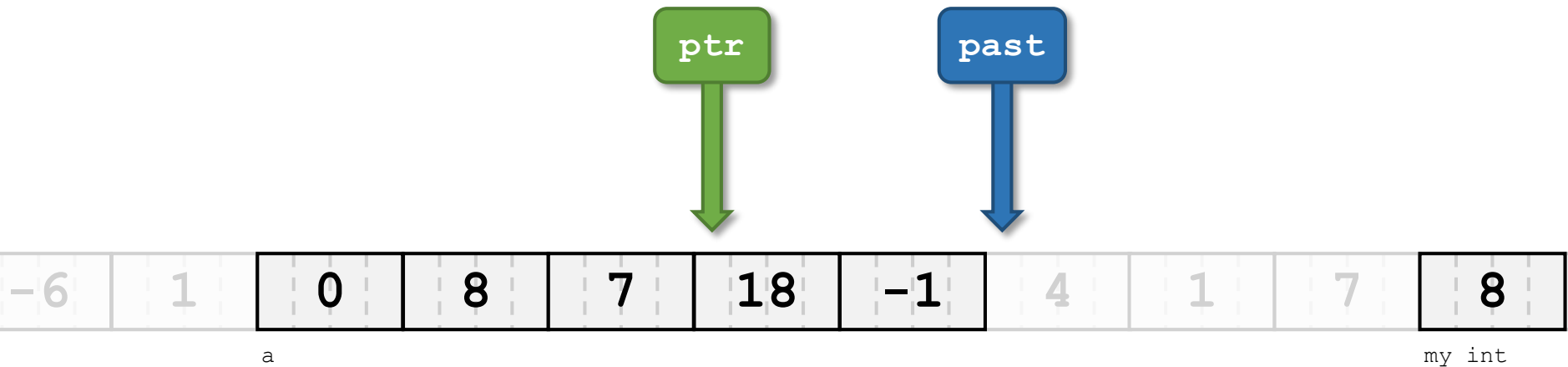
"To the left" means:

smaller index  
of element  
pointed to  
in array

Output: true

Because ptr is  
"to the left" of past.

```
int a[5] =  
int* ptr = a  
++ptr;  
int my_int = *ptr  
ptr += 2;  
*ptr = 18;  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```



# Program

"To the left" means:

smaller index  
of element  
pointed to  
in array

Output: true

Because ptr is  
"to the left" of past.

```
int a[5] =  
int* ptr = a  
++ptr;  
int my_int = *ptr  
ptr += 2;  
*ptr  
// arr  
// shi  
// read  
// shift  
// overw  
// compare pointers
```

Here:

Index 3 < Index 5



a

my\_int

# Pointer Program

```
int a[5] = {0, 8, 7, 2, -1};  
int* ptr = a; // array-to-pointer conv  
++ptr; // shift to the right  
int my_int = *ptr; // read target  
ptr += 2; // shift by 2 elements  
*ptr = 18; // overwrite target  
int* past = a+5;  
std::cout << (ptr < past) << "\n"; // compare pointers
```

