

Pointer Program

```
int* a = new int[5]{0, 8, 7, 2, -1};
int* ptr = a; // pointer assignment
++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

Find and fix at least 3 problems in the following program.

```
#include <iostream>
int main () {
    int* a = new int[7]{0, 6, 5, 3, 2, 4, 1};
    int* b = new int[7];
    int* c = b;
    // copy a into b using pointers
    for (int* p = a; p <= a+7; ++p) {
        *c++ = *p;
    }
    // cross-check with random access
    for (int i = 0; i <= 7; ++i) {
        if (a[i] != c[i]) {
            std::cout << "Oops, copy error...\n";
        }
    }
    return 0;
}
```

Pointer Program

```
#include <iostream>
int main () {
    int* a = new int[7]{0, 6, 5, 3, 2,
    int* b = new int[7];
    int* c = b;
    // copy a into b using pointers
    for (int* p = a; p <= a+7; ++p) {
        *c++ = *p;
    }
    // cross-check with random access
    for (int i = 0; i <= 7; ++i) {
        if (a[i] != c[i]) {
            std::cout << "Oops, copy error...\n";
        }
    }
    return 0;
}
```

`p = a+7` is dereferenced

Solution:

Use `<` instead of `<=`

Pointer Program

```
#include <iostream>
int main () {
    int* a = new int[7]{0, 6, 5, 3, 2, 1, 0};
    int* b = new int[7];
    int* c = b;
    // copy a into b using pointers
    for (int* p = a; p <= a+7; ++p) {
        *c++ = *p;
    }
    // cross-check with random access
    for (int i = 0; i <= 7; ++i) {
        if (a[i] != c[i]) {
            std::cout << "Oops, copy error\n";
        }
    }
    return 0;
}
```

`p = a+7` is dereferenced

Solution:

Use `<` instead of `<=`

Same problem as above

Pointer Program

```
#include <iostream>
int main () {
    int* a = new int[7]{0, 6, 5, 3, 2, 4, 1};
    int* b = new int[7];
    int* c = b;
    // copy a into b using pointers
    for (int* p = a; p <= a+7; ++p) {
        *c++ = *p;
    }
    // cross-check with random access
    for (int i = 0; i <= 7; ++i) {
        if (a[i] != c[i]) {
            std::cout << "Oops, copy error\n";
        }
    }
    return 0;
}
```

c doesn't point to b[0] anymore.

Solution:
Use b instead of c

p = a+7 is dereferenced

Solution:
Use < instead of <=

Same problem as above