

```
#include <iostream>
#include <algorithm>
#include <memory>
#include <vector>

using namespace std;

int main() {

    vector<unique_ptr<int>> intPointers;

    for (int i = 0; i < 10; i++) {
        unique_ptr tmp = make_unique<int>();
        *tmp = i;
        //duing example: remove the move, and show that it wont copy
        intPointers.push_back(std::move(tmp));
    }

    for_each(begin(intPointers), end(intPointers), [](const unique_ptr<int>& i) {
        cout << *i << " ";
    });

    cout << "\n";

    vector<unique_ptr<int>> morePointers;

    move(begin(intPointers), end(intPointers), back_inserter(morePointers));

    //After the move, those objects are now all nullptr
    cout << boolalpha;
    for_each(begin(intPointers), end(intPointers), [](const unique_ptr<int>& i) {
        cout << (i == nullptr) << " ";
    });

    cout << "\n";

    intPointers.clear(); //dont want to use them by accident!

    for_each(begin(morePointers), end(morePointers), [](const unique_ptr<int>& i) {
        cout << *i << " ";
    });

    cout << "\n";

    return 0;

}
```