- Exercise

In this exercise, you will implement one of the examples in the previous lesson with move semantic.

Task 1

In the program below, a `BigArray` with 10 billion entries will be pushed to an `std::vector`. Compile the program and measure its performance. The program requires that you compile this program for 64-bit.

Task 2

Extend `BigArray` with move semantic and measure the performance once more.

How big is the performance gain?

```cpp
#include <algorithm>
#include <chrono>
#include <iostream>
#include <vector>

using std::cout;
using std::endl;

using std::chrono::system_clock;
using std::chrono::duration;

using std::vector;

class BigArray{

public:
  BigArray(size_t len): len_(len), data_(new int[len]){}

  BigArray(const BigArray& other): len_(other.len_), data_(new int[other.len_]){
```

```cpp
        cout << "Copy construction of " << other.len_ << " elements "<< endl;
        std::copy(other.data_, other.data_ + len_, data_);
    }


    BigArray& operator=(const BigArray& other){
        cout << "Copy assignment of " << other.len_ << " elements "<< endl;
        if (this != &other){
            delete[] data_;

            len_ = other.len_;
            data_ = new int[len_];
            std::copy(other.data_, other.data_ + len_, data_);
        }
        return *this;
    }

    ~BigArray(){
        if (data_ != nullptr) delete[] data_;
    }

private:
    size_t len_;
    int* data_;
};

int main(){

    cout << endl;

    vector<BigArray> myVec;

    auto begin= system_clock::now();

    myVec.push_back(BigArray(1000000000));

    auto end= system_clock::now() - begin;
    auto timeInSeconds= duration<double>(end).count();

    cout << endl;
    cout << "time in seconds: " << timeInSeconds << endl;
    cout << endl;

}
```

Let's take a look at the solution in the next lesson.