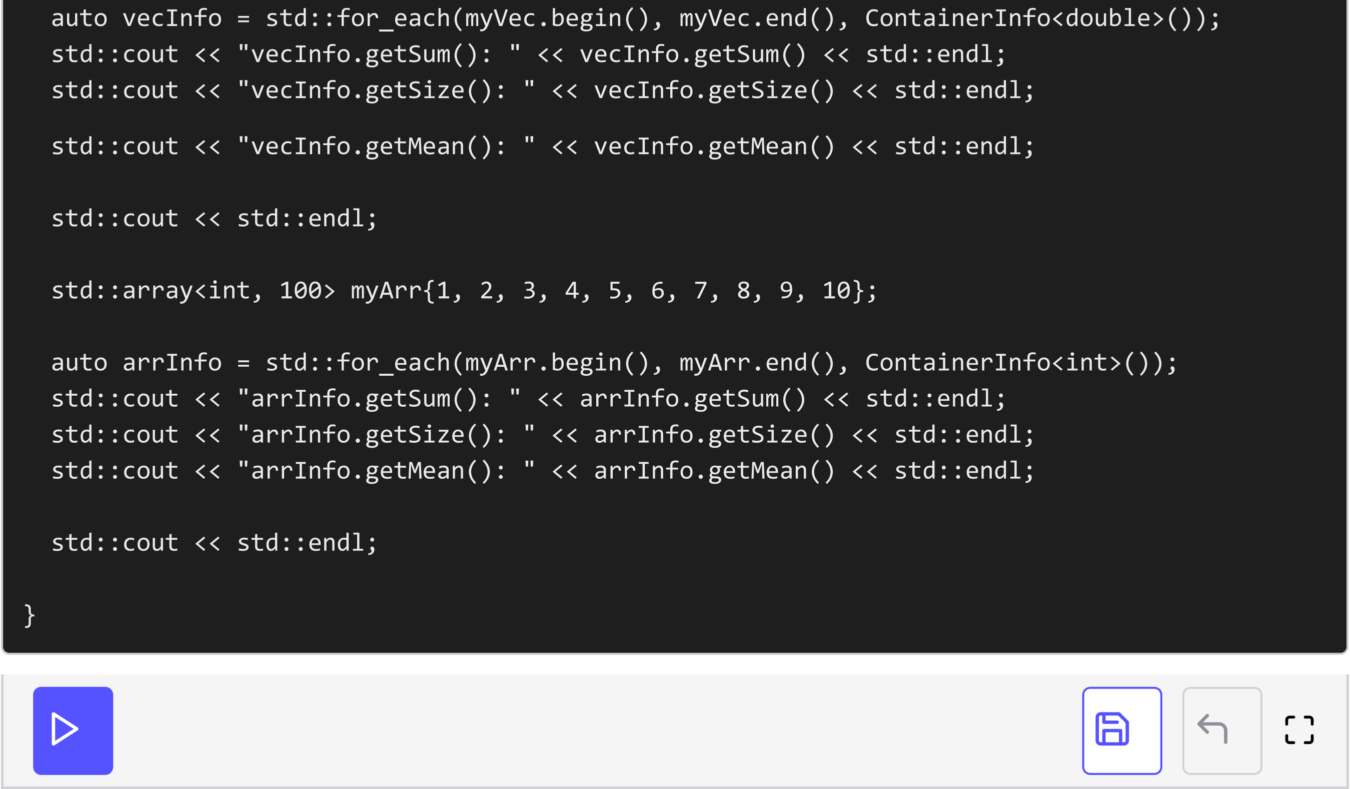
- Solution

Let's look at the solution of the exercise.

WE'LL COVER THE FOLLOWING

Solution

```
#include <array>
#include <algorithm>
#include <iostream>
#include <string>
#include <vector>

template <typename T>
class ContainerInfo{
public:

  void operator()(T t){
    size++;
    sum += t;
  }

  int getSum() const{
    return sum;
  }

  int getSize() const{ return size; }

  double getMean() const{
      return static_cast<double>(sum)/static_cast<double>(size);
  }
private:
  T sum{0};
  int size{0};
};

int main(){

  std::cout << std::endl;

  std::vector<double> myVec{1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9};
```



```
  auto vecInfo = std::for_each(myVec.begin(), myVec.end(), ContainerInfo<double>());
  std::cout << "vecInfo.getSum(): " << vecInfo.getSum() << std::endl;
  std::cout << "vecInfo.getSize(): " << vecInfo.getSize() << std::endl;
  std::cout << "vecInfo.getMean(): " << vecInfo.getMean() << std::endl;

  std::cout << std::endl;

  std::array<int, 100> myArr{1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

  auto arrInfo = std::for_each(myArr.begin(), myArr.end(), ContainerInfo<int>());
  std::cout << "arrInfo.getSum(): " << arrInfo.getSum() << std::endl;
  std::cout << "arrInfo.getSize(): " << arrInfo.getSize() << std::endl;
  std::cout << "arrInfo.getMean(): " << arrInfo.getMean() << std::endl;

  std::cout << std::endl;

}
```

Explanation

- We have created a vector, i.e., `myVec` and an array, i.e., `myArr` in lines 34 and 43.

- By using `for_each`, we have stored the information about the sum, size, and mean in the `ContainerInfo` by moving the iteration from beginning to end.

- `ContainerInfo` is stored for both the vector and array in `vecInfo` and `arrInfo`.

In the next chapter, we'll start off with non-modifying algorithms.