Removed std::random_shuffle

This lessons will provide a better alternative to random_shuffle.

The `random_shuffle(first, last)` and `random_shuffle(first, last, rng)` functions were marked already as deprecated in C++14. The reason was that in most cases it used the `rand()` function, which is considered inefficient and even error-prone (as it uses global state). Alternatively, you could provide the `rng` function parameter that appeared to be unusable in practice. If you need the same functionality, use `std::shuffle`:

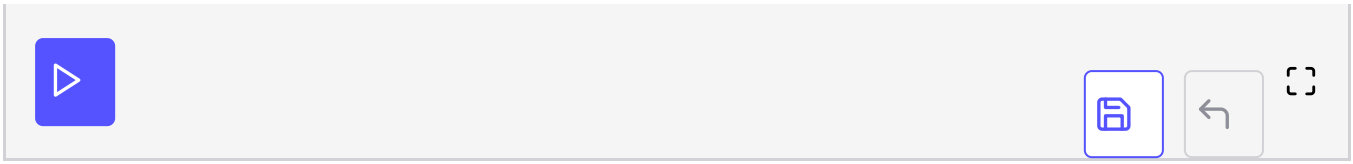
```
template< class RandomIt, class URBG >
void shuffle( RandomIt first, RandomIt last, URBG&& g );
```

`std::shuffle` takes a random number generator as the third template argument, which is safer, easier to use and more scalable.

Have a look at the following example on how to convert from `random_shuffle` to `shuffle`:

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

int main() {
    std::vector<int> vec = { 0, 1, 2, 3, 4, 5 };

    // Pre-C++17:
    std::random_shuffle(begin(vec), end(vec));

    for (auto& elem : vec)
        std::cout << elem << ", ";

    // C++17 version:
    std::random_device randDev;
    std::mt19937 gen(randDev());

    std::shuffle(begin(vec), end(vec), gen);

    for (auto& elem : vec)
        std::cout << elem << ", ";
}
```

}



> *Extra Info:* See more information in N4190.

In the next lesson, we will go over some old functional stuff that has been removed.