Move Ranges

In C++, we can move data from one range to another. Read the lesson for more details.

`std::move` moves the ranges forward; `std::move_backward` moves the ranges backwards.

`move`: moves the elements in the range `first` to `last` to the range starting from `result`.

```
OutIt move(InpIt first, InpIt last, OutIt result)
FwdIt2 move(ExePol pol, FwdIt first, FwdIt last, Fwd2It result)
```

`move_backward`: moves the elements in the range `first` to `last` to the range ending at `result`.

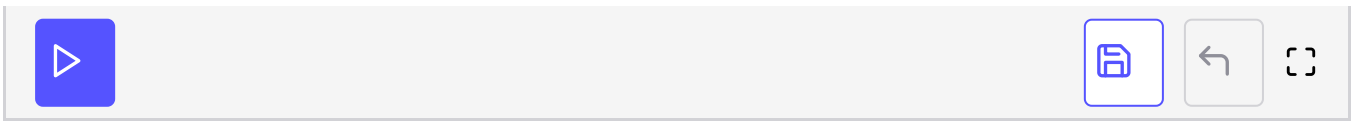
```
BiIt move_backward(BiIt first, BiIt last, BiIt result)
```

Both algorithms need a destination iterator `result`, to which the range is moved. In the case of the `std::move` algorithm this is an output iterator, and in the case of the `std::move_backward` algorithm this is a bidirectional iterator. The algorithms return an output or bidirectional iterator, pointing to the initial position in the destination range.

> ⚠ **The source range may be changed**
> `std::move` and `std::move_backward` use move semantics. Therefore the source range is valid, but doesn't necessarily have the same elements afterward.

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

```
int main(){

  std::cout << std::endl;


  std::vector<int> myVec1{0, 1, 2, 3, 4, 5, 6, 7, 9};
  std::vector<int> myVec2(10);

  std::move(myVec1.begin(), myVec1.end(), myVec2.begin());
  for ( auto v: myVec2 ) std::cout << v << " ";

  std::cout << "\n\n";

  std::string str1{"abcdefghijklmnop"};
  std::string str2{"--------------------"};

  std::cout << str2 << std::endl;
  std::move_backward(str1.begin(), str1.end(), str2.end());
  std::cout << str2 << std::endl;

  std::cout << std::endl;

}
```



Move ranges

In the next lesson, we'll discuss how we can swap data while moving ranges.