Copy Elements and Ranges

In this lesson, we'll learn how to perform various copy operations on a given range.

We can copy ranges forward with `std::copy`, backward with `std::copy_backward` and conditionally with `std::copy_if`. To copy n elements, we can use `std::copy_n`.

`copy`: copies the range.

```
OutIt copy(InpIt first, InpIt last, OutIt result)
FwdIt2 copy(ExePol pol, FwdIt first, FwdIt last, FowdIt2 result)
```

`copy_n`: copies n elements.

```
OutIt copy_n(InpIt first, Size n, OutIt result)
FwdIt2 copy_n(ExePol pol, FwdIt first, Size n, FwdIt2 result)
```

`copy_if`: Copies the elements dependent on the predicate `pre`.

```
OutIt copy_if(InpIt first, InpIt last, OutIt result, UnPre pre)
FwdIt2 copy_if(ExePol pol, FwdIt first, FwdIt last, FwdIt2 result, UnPre pre)
```

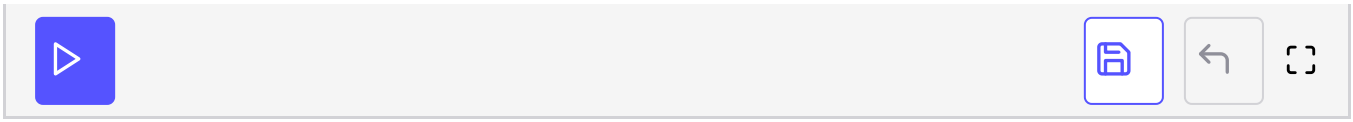
`copy_backward`: Copies the range backward.

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

The algorithms need input iterators to copy their elements to `result`. They return an end iterator to the destination range.

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

int main(){
```

```
  std::cout << std::endl;

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

  std::copy_if(myVec.begin(), myVec.end(), myVec2.begin()+3, [](int a){ return a%2; });
  for ( auto v: myVec2 ) std::cout << v << " ";

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

  std::string str{"Iamstring1"};
  std::string str2{"Hellostring----------------------2"};

  std::cout << str2 << std::endl;
  std::copy_backward(str.begin(), str.end(), str2.end());
  std::cout << str2 << std::endl;

  std::cout << std::endl;

  std::cout << str << std::endl;
  std::copy_backward(str.begin(), str.begin() + 5,  str.end());
  std::cout << str << std::endl;

  std::cout << std::endl;

}
```



Copy elements and ranges

Let’s look at the functions we can use to update and replace values in ranges in the next lesson.