Replace Elements and Ranges

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

You have with std::replace_copy and std::replace_copy_if four variations to replace elements in a range. The algorithms differ in two aspects. First, does the algorithm need a predicate? Second, does the algorithm copy the elements in the destination range?

replace: Replaces the old elements in the range with newValue, if the old element has the value old.

```
void replace(FwdIt first, FwdIt last, const T& old, const T& newValue)
void replace(ExePol pol, FwdIt first, FwdIt last, const T& old, const T& newValue)
```

replace_if: Replaces the old elements of the range with newValue, if the old
element fulfils the predicate pred:

```
void replace_if(FwdIt first, FwdIt last, UnPred pred, const T& newValue)
void replace_if(ExePol pol, FwdIt first, FwdIt last, UnPred pred, const T& newValue)
```

replace_copy : Replaces the old elements in the range with newValue , if the old
element has the value old . Copies the result to result :

```
OutIt replace_copy(InpIt first, InpIt last, OutIt result, const T& old, const T& newValue)
FwdIt2 replace_copy(ExePol pol, FwdIt first, FwdIt last, FwdIt2 result, const T& old, const T
```

replace_copy_if : Replaces the old elements of the range with newValue, if the
old element fulfills the predicate pred . Copies the result to result :

```
OutIt replace_copy_if(InpIt first, InpIt last, OutIt result, UnPre pred, const T& newValue)
FwdIt2 replace_copy_if(ExePol pol, FwdIt first, FwdIt last, FwdIt2 result, UnPre pred, const
```

The algorithms in action:

```
#include <algorithm>
#include <cctype>
#include <iostream>
#include <string>
int main(){
  std::cout << std::endl;</pre>
  std::string str{"Only for testing purpose."};
  std::cout << str << std::endl;</pre>
  std::replace(str.begin(), str.end(), ' ', '1');
  std::cout << str << std::endl;</pre>
  std::replace_if(str.begin(), str.end(), [](char c){ return c == '1'; }, '2');
  std::cout << str << std::endl;</pre>
  std::string str2;
  std::replace_copy(str.begin(), str.end(), std::back_inserter(str2), '2', '3');
  std::cout << str2 << std::endl;</pre>
  std::string str3;
  std::replace_copy_if(str2.begin(), str2.end(), std::back_inserter(str3), [](char c){ return
  std::cout << str3 << std::endl;</pre>
  std::cout << std::endl;</pre>
}
```







ני

Replace elements and ranges