remove(p, q, x);
// removes all occurrences of x from [p, q[, shifting (copying) the
// remaining elements to the left;
// invariant: the length of the segment remains unchanged;

EXAMPLE E.42 Testing the remove() Algorithm

```cpp
int main()
{ char* s="All is flux, nothing is stationary."; // Heraclitus
  int l = strlen(s);
  int n = count(s, s+l, ' ');
  cout << "l=" << l << " \n";
  cout << "n=" << n << " \n";
  remove(s, s+l, ' ');
  cout << s << " \n";
  s[l-n] = 0; // truncate s
  cout << s << " \n";
}
```

l=35
n=5
All is flux, nothing is stationary.
All is flux, nothing is stationary.

Since 5 blanks were removed, the last 5 letters remain after their copies were shifted left.

remove_copy(p, q, pp, x);
// copies all elements of [p, q[ that do not match x to [pp, pp+n[, where n is the number of nonmatching elements;
// returns pp+n;
// invariant: [p, q[ remains unchanged;

EXAMPLE E.43 Testing the remove_copy() Algorithm

```cpp
int main()
{ char* s="All is flux, nothing is stationary."; // Heraclitus
  char buffer[80];
  int l = strlen(s);
  int n = count(s, s+l, ' ');
  cout << "l=" << l << " \n";
  cout << "n=" << n << " \n";
  char* ss = remove_copy(s, s+l, buffer, ' ');
  *ss = 0; // truncate buffer
  cout << s << " \n";
  cout << buffer << " \n";
  cout << ss-buffer << " \n";
}
```

l=35
n=5
All is flux, nothing is stationary.
All is flux, nothing is stationary.
30
remove_copy_if(p, q, pp, P());
// copies all elements x of [p,q[ for which !P(x) to [pp,pp+n[, 
// where n is the number of nonmatching elements;
// returns pp+n;
// invariant: [p,q[ remains unchanged;

EXAMPLE E.44 Testing the remove_copy_if() Algorithm

class Blank
{ public:
   bool operator()(char c) { return c == ' '; }
};

int main()
{ char* s="All is flux, nothing is stationary."; // Heraclitus
   char buffer[80];
   int l = strlen(s);
   int n = count(s, s+l, ' ');
   cout << "l=" << l << endl;
   cout << "n=" << n << endl;
   char* ss = remove_copy_if(s, s+l, buffer, Blank());
   *ss = 0; // truncate buffer
   cout << s << endl;
   cout << buffer << endl;
   cout << ss-buffer << endl;
}  
l=35
n=5
All is flux, nothing is stationary.
Allisflux,nothingisstationary.
30

This is the same as Example E.43 except that a predicate is used.

remove_if(p, q, P());
// removes all x from [p,q[ for which !P(x), shifting (copying) the 
// remaining elements to the left;

EXAMPLE E.45 Testing the remove_if() Algorithm

class Blank
{ public:
   bool operator()(char c) { return c == ' '; }
};

int main()
{ char* s="All is flux, nothing is stationary."; // Heraclitus
   int l = strlen(s);
   int n = count(s, s+l, ' ');
   cout << "l=" << l << endl;
   cout << "n=" << n << endl;
   remove_if(s, s+l, Blank());
   cout << "s=" << s << endl;
   s[l-n] = 0;
}