```
        print(b,3);
    }
    n=10: {100,111,122,133,144,155,166,177,188,199}
    n=10: {100,111,177,188,199,155,166,177,188,199}
    n=3:  {177,188,199}
```

count(p,q,x);
// returns the number of occurrences of x in the segment [p,q[;
// invariant: [p,q[is left unchanged;

EXAMPLE E.7 Testing the `count()` Algorithm

```
    int main()
    { int a[] = {0,1,0,1,1,1,0,1,1,0};
      print(a,10);
      int n = count(a,a+10,1);
      cout << "n=" << n << '\n';
    }
    n=10: {0,1,0,1,1,1,0,1,1,0}
    n=6
```

count_if(p,q,P());
// returns the number of occurrences where P(x) in the segment [p,q[;
// invariant: [p,q[is left unchanged;

EXAMPLE E.8 Testing the `count_if()` Algorithm

```
    int main()
    { int a[] = {0,1,0,1,1,1,0,1,1,0};
      print(a,10);
      int n = count_if(a,a+10,Odd());
      cout << "n=" << n << '\n';
    }
    n=10: {0,1,0,1,1,1,0,1,1,0}
    n=6
```

equal(p,q,pp);
// returns true iff the segment [p,q) matches [pp,pp+n[, where n = q-p;
// invariant: [p,q[and [pp,qq+n[are left unchanged;

EXAMPLE E.9 Testing the `equal()` Algorithm

```
    int main()
    { int a[] = {0,1,0,1,1,1,0,1,1,0};
      int b[] = {0,1,0,0,1,1,0,1,0,0};
      print(a,10);
      print(b,10);
      cout << "equal(a,a+10,b)=" << equal(a,a+10,b) << '\n';
      cout << "equal(a+1,a+4,a+5)=" << equal(a+1,a+4,a+5) << '\n';
```