```
=== 演習1 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-1.cpp
$ ./a.out
2
count(2): 3
1
count(1): 2
3
count(3): 2
5
count(5): 0
4
count(4): 1


--- ex09-1.cpp ---
// count
#include <algorithm>
#include <iostream>
#include <vector>

int main()
{
   std::vector a {1,2,3,1,2,2,3,4};
   for (int x{}; std::cin >> x; ) {
      std::cout <<"count("<< x <<"): "
        << std::count(a.begin(), a.end(), x)
        << "\n";
   }
}
```

```
=== 演習2 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-2.cpp
$ ./a.out
10
10 => (2)(5):2
12
12 => (2)(3):2
33
33 => (3)(11):2
21
21 => (3)(7):2


--- ex09-2.cpp ---
// count_if
#include <algorithm>
#include <iostream>
#include <vector>

int input;
bool pred(int x) {
   bool a = input % x == 0;
   if (a) std::cout <<"("<< x <<")";
   return a;
}

int main()
{
   std::vector a {2,3,5,7,11};
   while (std::cin >> input) {
      std::cout << input <<" => ";
      auto x { std::count_if(a.begin(), a.end(), pred) };
      std::cout <<":" << x << "\n";
   }
}
```

```
=== 演習3 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-3.cpp
$ ./a.out
trimmean = 4.75


--- ex09-3.cpp ---
// count_if
#include <algorithm>
#include <iostream>
#include <deque>

int main()
{
   std::deque d {4,2,1,5,9,8,9};
   auto [min,max] { std::minmax_element(d.begin(), d.end()) };
   double sum{0};
   int num{0};
   for (auto e:d) {
      if (e != *min && e != *max) {
         sum += e;
         ++ num;
      }
   }
   if (num)
      std::cout <<"trimmean = "<< sum/num <<"\n";

}
```

```
=== 演習4 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-4.cpp
$ ./a.out
17


--- ex09-4.cpp ---
// std::find_if_not
#include <algorithm>
#include <iostream>
#include <vector>

bool pred(int x) { return x % 2 == 0 || x % 3 == 0; }

int main()
{
   std::vector a {2,6,18,17,12,3};
   auto it { std::find_if_not(a.begin(),
                              a.end(), pred) };
   std::cout << *it <<"\n";
}
```

```
=== 演習5 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-5.cpp
$ ./a.out
usage:./a.out count value
$ ./a.out 2 3
found at 2
$ ./a.out 2 4
found at 4
$ ./a.out 5 2
not found
$ ./a.out 4 2
found at 6
$ ./a.out 3 2
found at 6


--- ex09-5.cpp ---
// search_n
#include <algorithm>
#include <iostream>
#include <vector>

using std::cout;

int main(int argc, char *argv[])
{
   if (argc < 3) {
      cout <<"usage:"<< argv[0]<<" count value\n";
      return 1;
   }
   int c{std::stoi(argv[1])}, v{std::stoi(argv[2])};
   std::vector a {1,4,3,3,4,4,2,2,2,2,3};
   auto it {std::search_n(a.begin(), a.end(), c, v)};
   if (it == a.end())
      cout <<"not found\n";
   else
      cout <<"found at "<< it-a.begin() <<"\n";
}
```

```
=== 演習6 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-5.cpp
$ ./a.out
3 4 7


--- ex09-5.cpp ---
// std::search
#include <algorithm>
#include <iostream>
#include <vector>

bool pred(int x, int y) { return std::abs(x-y) % 2 == 0; }

int main()
{
   std::vector a{2,3,4,7,8,1,2,3,4}, b{1,2,3};
   auto it { std::search(a.begin(), a.end(),
                         b.begin(), b.end(), pred) };
   if (it != a.end()) {
      auto idx {it-a.begin()};
      for (size_t i = 0; i < b.size(); ++i)
         std::cout << a[idx+i] <<" ";
      std::cout <<"\n";
   }
}
```

```
=== 演習7 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex09-7.cpp
$ ./a.out
found at 9


--- ex09-7.cpp ---
// implemenation of std::find_end
#include <algorithm>
#include <iostream>
#include <vector>

// 他にもいろいろな実装方法があります
template<typename T, typename K>
T myfind_end(T b1, T e1, K b2, K e2)
{
   T ita { std::search(b1, e1, b2, e2) };
   while (ita != e1) {
      auto itb { std::search(std::next(ita), e1, b2, e2) };
      if (itb == e1) return ita;
      ita = itb;
   }
   return ita;
}


int main()
{
   std::vector a{3,1,2,3,8,1,2,3,5,1,2,3,6}, s{1,2,3};
   auto it { myfind_end(a.begin(), a.end(), s.begin(), s.end()) };
   if (it != a.end())
      std::cout <<"found at "<< it-a.begin() <<"\n"; // 9
}
```