

## 第7回演習

# 演習 1

- ▶ 資料 p.10 を参考にコンテナの中身を出力する print() 関数を書け。範囲 for 文は用いない。
- ▶ vector, deque, list, set に対応できること
- ▶ ヒント：先頭要素以外はまず", "を出力する

```
template<typename T> // ex07-1.cpp
void print(const T& x) {
    // ここを考える
}

int main() {
    std::vector a {1,2,3,4,5};
    std::deque b {3,5,4,6,8};
    std::list c {7,4,2,1,9};
    std::set d {6,2,8,6,2};
    print(a); // 1, 2, 3, 4, 5
    print(b); // 3, 5, 4, 6, 8
    print(c); // 7, 4, 2, 1, 9
    print(d); // 2, 6, 8
}
```

## 演習 2

- ▶ 演習 1 で作った print 関数を ex07print.hpp にいれよ
- ▶ 資料 p.14 の find と insert の組み合わせをためすテンプレート find\_and\_insert 関数を作れ。
- ▶ 第 1 引数をコンテナ変数、第 2 引数を探す数、第 3 引数に挿入する数とする。

```
// 必要なヘッダファイルを書く,    ex07-2.cpp
#include "ex07print.hpp"
template<typename T>
void find_and_insert(T& c, int f, int in) {
    // p.14 を参考にする
}

int main() {
    std::list x {3,5,2,8,9,6,4};
    find_and_insert(x, 9, 10); // 9
    print(x); // 3, 5, 2, 8, 10, 9, 6, 4
    std::vector y {3,5,2,8,9,6,4};
    find_and_insert(y, 6, 7); // 6
    print(y); // 3, 5, 2, 8, 9, 7, 6, 4
}
```

## 演習 3

- ▶ 資料 p.15 のプログラムを main 関数で実行して出力を確認せよ。

```
#include <vector>    // ex07-3
#include "ex07print.hpp"

int main() {
    std::vector x {10,3,10,8,10,10,4};
    // すべての 10 の前に 5 を挿入
    for (auto it = x.begin(); it != x.end(); ++it) {
        if (*it == 10) {
            it = x.insert(it, 5); // 5 のイテレータ
            ++ it;                // 10 のイテレータ
        }
    }
    print(x);
    // 5, 10, 3, 5, 10, 8, 5, 10, 5, 10, 4
}
```

## 演習 4

- ▶ 資料 p.16 のプログラムを main 関数で実行して出力を確認せよ。

```
#include <vector>    // ex07-4.cpp
#include "ex07print.hpp"

int main() {
    std::vector x {3,5,3,8,3,6,3};
    // すべての3を削除
    auto it { x.begin() };
    while (it != x.end()) {
        if (*it == 3)
            it = x.erase(it);
        else
            ++it;
    }

    print(x); // 5, 8, 6
}
```

## 演習 5

- ▶ 資料 p.17 のプログラムを main 関数で実行して出力を確認せよ。

```
#include <iostream> // ex07-5.cpp
#include <map>
int main() {
    std::multimap<std::string,int> x
    { {"xn",10}, {"ya",5}, {"xn", 5}, {"zn",3},
      {"xn",36}, {"sa",5}, {"xn",24}, {"tn",3}};

    auto r { x.equal_range("xn") };
    for (auto it = r.first; it != r.second; ++it) {
        auto [k,v] {*it};
        std::cout << "[" << k << ", " << v << "]" << " ";
    }
    std::cout << "\n"; // [xn,10] [xn,5] [xn,36] [xn,24]
}
```

## 演習 6

- ▶ 資料 p.18 のプログラムを main 関数で実行して出力を確認せよ。

```
#include <iostream> // ex07-6.cpp
#include <map>

int main() {
    std::map<std::string,int> x
    { {"e",10}, {"a",5}, {"i", 5}, {"h",3},
      {"f",36}, {"b",5}, {"d",24}, {"g",3}};

    auto l {x.lower_bound("c")}; // "c"はない
    auto u {x.upper_bound("f")};
    for (auto it = l; it != u; ++it) {
        auto [k,v] {*it};
        std::cout << "[" << k << ", " << v << "] ";
    }
    std::cout << "\n"; // [d,24] [e,10] [f,36]
}
```

## 演習 7

- ▶ 資料 p.25 のプログラムを `std::list` で実行して出力を確認せよ。

```
#include <iostream>
#include <iterator>
#include <list>
int main()
{
    std::list x {1,2,3,4,5,6,7,8,9,10};
    for (auto it {x.begin()};
         it !=x.end(); std::advance(it, 2)) {
        std::cout << std::distance(x.begin(), it)
                  << ":" << *it << " ";
    }
    std::cout << "\n"; // 0:1 2:3 4:5 6:7 8:9
}
```



## 演習 8

- ▶ 資料 p.31, p.32 を参考にして、標準入力から入ってきた整数の列を 2 倍して出力するプログラムを作成せよ。
- ▶ >>演算子や<<演算子を使用してはならない。

```
#include <iostream> // ex07-8.cpp
#include <iterator>

int main() {
    std::istream_iterator<int> in{std::cin};
    std::istream_iterator<int> eos;
    std::ostream_iterator<int> out{std::cout, ", "};
    // ここを埋める
    //
}
```

```
$ echo 1 2 3 4 5 | ./a.out
2, 4, 6, 8, 10,
```