

## 第 6 回 C++プログラミング実験 II

出題日:2022 年 5 月 16 日(月)  
提出期限:2022 年 5 月 19 日(木)24:00

### 実験課題 6-1

CoursePower からスケルトンファイル ex06-1-main.cpp をダウンロードし、以下の問題に解答せよ。

std::list にはメンバ関数として、要素を削除する機能 erase() がある。実際に、ダウンロードした ex06-1-main.cpp の 27 行目から 33 行目ではこのメンバ関数 erase() を使って先頭に一番近い 3 を削除している。

この問題では、ex06-1-main.cpp の既に記述されている部分や下記のヒントを参考にしながら、メンバ関数としての erase() を用いずに、「あるリスト lst に対し、指定された値 n を持つ要素の中で先頭に一番近いもののみを削除したリストを新たに作成し、それを lst に代入することで lst を更新させる」ような erase 関数を一般関数として作成し、ex06-1-main.cpp を完成させよ(「#####」の箇所のみを記述するだけで完成できる)。ただし、この一般関数の erase 関数において、要素に存在しない値が指定されたときは元のリストは変化しない。また、要素を削除し更新されたリストにおいては、他の要素の置かれている場所情報は変わっても構わない(メンバ関数と機能が異なってもよい)。

ヒント 1: リスト L 内の要素は for(auto& e :L) を用いることで先頭から順に探索できる。

ヒント 2: std::list のメンバ関数 splice() を用いて、2 つのリスト L1, L2 に対し、L1.splice(L1.end(), L2) とすれば、L1 の末尾に L2 が結合される。

#### 実行結果1

```
$ g++ -std=c++17 ex06-1-main.cpp -o ex06-1
$ ./ex06-1
The original list is as follows:
L0 = 1 3 8 4 5 3 5 9

By finding '3' from the original list L0 and erasing it,
L1 = 1 8 4 5 3 5 9

Select a number from the list L1 to erase: 1 (←自分で入力)

By finding '1' from the list L1 and erasing it,
L2 = 8 4 5 3 5 9
```

#### 実行結果2

```
$ g++ -std=c++17 ex06-1-main.cpp -o ex06-1
$ ./ex06-1
The original list is as follows:
L0 = 1 3 8 4 5 3 5 9

By finding '3' from the original list L0 and erasing it,
L1 = 1 8 4 5 3 5 9

Select a number from the list L1 to erase: 5 (←自分で入力)

By finding '5' from the list L1 and erasing it,
L2 = 1 8 4 3 5 9
```

※問題文下線部に関する注意：一般関数の erase 関数をうまく書かなければ、実行結果2のように標準入力すると、

By finding '5' from the list L1 and erasing it,

L2 = 1 8 4 3 9

というように、リスト内にあるすべての n (=5) が削除されてしまう。

## 第 6 回 C++プログラミング実験 II

出題日:2022 年 5 月 16 日(月)  
提出期限:2022 年 5 月 19 日(木) 24:00

### 実験課題 6-2

仮定の動物園にいる動物を `std::set<string> zoo` にて管理することを考える。まず既に在籍している, `monkey`, `elephant`, `rabbit`, `panda`, `bear` を `zoo` へ追加していく(9~13 行目)。次に, `find()`を利用して, `std::vector<string> search` の要素が `zoo` にあるかを確認する(空欄(1))。最後に, 標準入力から新たな動物を `zoo` へ追加し, 追加できたのか, すでにあつたために追加できなかったのかを標準出力する(空欄(2))。講義資料にある通り, `set` では同一キーがすでにある場合, 追加することはできない。

実行例を参考にしながら, `ex06-2-main.cpp` の空欄箇所に必要なコードを追記せよ。

#### ex06-2-main.cpp

```
#include<iostream>
#include<vector>
#include<set>
using std::cin,std::cout,std::string,std::set,std::vector;

int main(){
    set<string> zoo;

    zoo.insert( "monkey" );
    zoo.insert( "elephant" );
    zoo.insert( "rabbit" );
    zoo.insert( "panda" );
    zoo.insert( "bear" );

    vector<string> search{ "giraffe","panda"};

    //find()を利用して, zoo の中に search の要素があるか確認.
    for(int i=0; i<search.size(); i++){
        auto it { /*空欄(1) ここに必要なコードを記述*/ };
        if(it==zoo.end()) cout<< search[ i ] <<" is not found.\n";
        else
            cout<<search[ i ]<<" is found."<<"\n";
    }
    //標準入力から新しい動物を追加し, 追加できたのか, すでにあつたのかを出力する.
    /* 空欄(2) ここに必要なコードを記述 */

}
```

#### 実行結果

下線部は入力箇所

```
$ g++ -std=c++17 ex06-2-main.cpp -o ex06-2
$ ./ex06-2
giraffe is not found.
panda is found.
Input newcomer: giraffe
giraffe is inserted
Input newcomer: rabbit
rabbit has been already inserted.
Input newcomer: Ctrl+c
```

## 第 6 回 C++プログラミング実験 II

出題日:2022 年 5 月 16 日(月)  
提出期限:2022 年 5 月 19 日(木)24:00

### 実験課題 6-3

5 人の投票者が候補者の名前を入力するので、実行例のように

- 投票によって最大得票数を得た人物の名前
- 特定の人物 (Alice) の得票数

を表示するよう、以下のプログラムを完成させよ。

なお、得票数が最も多い人物が複数いる場合は全ての名前を出力すること。

#### ex06-3-main.cpp

```
#include <iostream>
#include <string>
#include <map>
using std::cout, std::cin, std::string, std::map;

int main(){
    map<string, int> cand;

    // 候補者名の入力
    for (int i = 0; i < 5; i++) {

    }

    // 最大得票者の出力

    // Alice の得票数の出力

    return 0;
}
```

#### 実行結果 1

```
$ g++ -std=c++17 ex06-3-main.cpp -o ex06-3
$ ./ex06-3
Name? Alice
Name? Bob
Name? Alice
Name? Alice
Name? Bob
Alice
Alice: 3
```

#### 実行結果 2

```
$ g++ -std=c++17 ex06-3-main.cpp -o ex06-3
$ ./ex06-3
Name? Alice
Name? Bob
Name? Alice
Name? Bob
Name? Carol
Alice
Bob
Alice: 2
```