

🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 081 - 템플릿 함수

개요	제출	편집	코딩 결과
----	----	----	-------

## 실습 081 - 템플릿 함수

제출 마감일: 2023-05-14 23:59

업로드 가능한 파일 수: 2

제출 방식: 개인

### 목적

이 실습은 template 함수를 정의하는 연습을 합니다.

### 설명

우리는 STL 에서 제공하는 containers 에 저장된 원소에 접근하기 위해서 std::iterator 를 사용합니다.

이터레이터를 이용하면 containers 의 종류에 상관 없이 알고리즘 함수를 이용해 합계를 구할 수 있습니다.

예를 들어, `std::vector<int> vec = {1, 2, 3, 4, 5};`

`int sum = std::accumulate(std::begin(vec), std::end(vec), 0);`

### 문제

std::iterator 를 이용해 STL containers 의 원소의 합을 구하는 my\_accumulate() 템플릿 함수를 정의하시오.

### 입력

없음

### 출력

```
1 2 3 4 5
15
1 2 3 4 5
15
```

### 제출파일

81.csv

my\_accumulate.h

## 참고

// main.cpp

```
#include "my_accumulate.h"
#include <cassert>
#include <numeric>
#include <iostream>
#include <vector>
#include <list>

int main() {
    std::vector<int> vec = {1, 2, 3, 4, 5};

    for (const auto& it : vec)
        std::cout << it << ' ';
    std::cout << '\n';
    std::cout << my_accumulate(vec.cbegin(), vec.cend(), 0) << std::endl;
    assert (my_accumulate(vec.cbegin(), vec.cend(), 0) == accumulate(vec.cbegin(), vec.cend(), 0));

    std::list<int> numbers = {1, 2, 3, 4, 5};
    for (const auto& it : numbers)
        std::cout << it << ' ';
    std::cout << '\n';
    std::cout << my_accumulate(numbers.cbegin(), numbers.cend(), 0) << std::endl;
    assert (my_accumulate(numbers.cbegin(), numbers.cend(), 0) == accumulate(numbers.cbegin(), numbers.cend(), 0));

    return 0;
}
```

// my\_accumulate.h

```
template<typename T, typename U>
U my_accumulate(T _begin, T _end, U init){
    //템플릿 함수를 구현하시오
}
```