

```

/* The following code example is taken from the book
 * "The C++ Standard Library – A Tutorial and Reference, 2nd Edition"
 * by Nicolai M. Josuttis, Addison-Wesley, 2012
 *
 * (C) Copyright Nicolai M. Josuttis 2012.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#ifndef ALGOSTUFF_HPP
#define ALGOSTUFF_HPP

#include <array>
#include <vector>
#include <deque>
#include <list>
#include <forward_list>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iterator>
#include <functional>
#include <numeric>
#include <iostream>
#include <string>

// INSERT_ELEMENTS (collection, first, last)
// - fill values from first to last into the collection
// - NOTE: NO half-open range
template <typename T>
inline void INSERT_ELEMENTS (T& coll, int first, int last)
{
    for (int i=first; i<=last; ++i) {
        coll.insert(coll.end(),i);
    }
}

// PRINT_ELEMENTS()
// - prints optional string optcstr followed by
// - all elements of the collection coll
// - separated by spaces
template <typename T>
inline void PRINT_ELEMENTS (const T& coll,
                           const std::string& optcstr="")
{
    std::cout << optcstr;
    for (auto elem : coll) {
        std::cout << elem << ' ';
    }
    std::cout << std::endl;
}

// PRINT_MAPPED_ELEMENTS()

```

```

// - prints optional string optcstr followed by
// - all elements of the key/value collection coll
// - separated by spaces
template <typename T>
inline void PRINT_MAPPED_ELEMENTS (const T& coll,
                                   const std::string& optcstr="")
{
    std::cout << optcstr;
    for (auto elem : coll) {
        std::cout << '[' << elem.first
                    << ',' << elem.second << "]" ";
    }
    std::cout << std::endl;
}

#endif /*ALGOSTUFF_HPP*/

```