

```

/* 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.
 */
#include <tuple>
#include <iostream>

// helper: print element with index IDX of tuple with MAX elements
template <int IDX, int MAX, typename... Args>
struct PRINT_TUPLE {
    static void print (std::ostream& strm, const std::tuple<Args...>& t) {
        strm << std::get<IDX>(t) << (IDX+1==MAX ? "" : ",");
        PRINT_TUPLE<IDX+1, MAX, Args...>::print(strm, t);
    }
};

// partial specialization to end the recursion
template <int MAX, typename... Args>
struct PRINT_TUPLE<MAX, MAX, Args...> {
    static void print (std::ostream& strm, const std::tuple<Args...>& t) {
    }
};

// output operator for tuples
template <typename... Args>
std::ostream& operator << (std::ostream& strm,
                           const std::tuple<Args...>& t)
{
    strm << "[";
    PRINT_TUPLE<0, sizeof... (Args), Args...>::print(strm, t);
    return strm << "]";
}

```