

```

/* 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 <iostream>
#include <list>
#include <algorithm>
#include <iterator>
#include "print.hpp"
using namespace std;

class IntSequence {
private:
    int value;
public:
    // constructor
    IntSequence (int initialValue)
        : value(initialValue) {

    // "function call"
    int operator() () {
        return value++;
    }
};

int main()
{
    list<int> coll;
    IntSequence seq(1);    // integral sequence starting with 1

    // insert values from 1 to 4
    // - pass function object by reference
    //   so that it will continue with 5
    generate_n<back_inserter_iterator<list<int>>,
                int, IntSequence&>(back_inserter(coll),    // start
                                   4,    // number of elements
                                   seq);    // generates values

    PRINT_ELEMENTS(coll);

    // insert values from 42 to 45
    generate_n (back_inserter(coll),    // start
                4,    // number of elements
                IntSequence(42));    // generates values
    PRINT_ELEMENTS(coll);

    // continue with first sequence
    // - pass function object by value
    //   so that it will continue with 5 again
    generate_n (back_inserter(coll),    // start

```

```

        4,                // number of elements
        seq);            // generates values
PRINT_ELEMENTS(coll);

// continue with first sequence again
generate_n (back_inserter(coll), // start
        4,                // number of elements
        seq);            // generates values
PRINT_ELEMENTS(coll);
}

```