

```

/* The following code example is taken from the book
 * "The C++ Standard Library – A Tutorial and Reference"
 * by Nicolai M. Josuttis, Addison-Wesley, 1999
 *
 * (C) Copyright Nicolai M. Josuttis 1999.
 * 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 <valarray>
using namespace std;

// print three-dimensional valarray line-by-line
template<class T>
void printValarray3D (const valarray<T>& va, int dim1, int dim2)
{
    for (int i=0; i<va.size()/(dim1*dim2); ++i) {
        for (int j=0; j<dim2; ++j) {
            for (int k=0; k<dim1; ++k) {
                cout << va[i*dim1*dim2+j*dim1+k] << ' ';
            }
            cout << '\n';
        }
        cout << '\n';
    }
    cout << endl;
}

int main()
{
    /* valarray with 24 elements
     * - two groups
     * - four rows
     * - three columns
     */
    valarray<double> va(24);

    // fill valarray with values
    for (int i=0; i<24; i++) {
        va[i] = i;
    }

    // print valarray
    printValarray3D (va, 3, 4);

    // we need two two-dimensional subsets of three times 3 values
    // in two 12-element arrays
    size_t lengthvalues[] = { 2, 3 };
    size_t stridevalues[] = { 12, 3 };
    valarray<size_t> length(lengthvalues, 2);
    valarray<size_t> stride(stridevalues, 2);

    // assign the second column of the first three rows
    // to the first column of the first three rows

```

```

va[gslice(0, length, stride)]
    = valarray<double>(va[gslice(1, length, stride)]);

// add and assign the third of the first three rows
// to the first of the first three rows
va[gslice(0, length, stride)]
    += valarray<double>(va[gslice(2, length, stride)]);

// print valarray
printValarray3D (va, 3, 4);
}

```