

```

/* 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 <random>
#include <map>
#include <string>
#include <iostream>
#include <iomanip>

template <typename Distr, typename Eng>
void distr (Distr d, Eng e, const std::string& name)
{
    // print min, max and four example values
    std::cout << name << ":" << std::endl;
    std::cout << "- min(): " << d.min() << std::endl;
    std::cout << "- max(): " << d.max() << std::endl;
    std::cout << "- values: " << d(e) << ', ' << d(e) << ', '
        << d(e) << ', ' << d(e) << std::endl;

    // count the generated values (converted to integral values)
    std::map<long long,int> valuecounter;
    for (int i=0; i<200000; ++i) {
        valuecounter[d(e)]++;
    }

    // and print the resulting distribution
    std::cout << "====" << std::endl;
    for (auto elem : valuecounter) {
        std::cout << std::setw(3) << elem.first << ": "
            << elem.second << std::endl;
    }
    std::cout << "====" << std::endl;
    std::cout << std::endl;
}

int main()
{
    std::knuth_b e;

    std::uniform_real_distribution<> ud(0, 10);
    distr(ud, e, "uniform_real_distribution");

    std::normal_distribution<> nd;
    distr(nd, e, "normal_distribution");

    std::exponential_distribution<> ed;
    distr(ed, e, "exponential_distribution");

    std::gamma_distribution<> gd;

```

```
    distr(gd, e, "gamma_distribution");  
}
```