```
/* 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 <future>
#include <thread>
#include <iostream>
#include <exception>
#include <stdexcept>
using namespace std;

int queryNumber ()
{
    // read number
    cout << "read number: ";
    int num;
    cin >> num;

    // throw exception if none
    if (!cin) {
        throw runtime_error("no number read");
    }

    return num;
}

void doSomething (char c, shared_future<int> f)
{
    try {
        // wait for number of characters to print
        int num = f.get();  // get result of queryNumber()

        for (int i=0; i<num; ++i) {
            this_thread::sleep_for(chrono::milliseconds(100));
            cout.put(c).flush();
        }
    }
    catch (const exception& e) {
        cerr << "EXCEPTION in thread " << this_thread::get_id()
                  << ": " << e.what() << endl;
    }
}

int main()
{
    try {
        // start one thread to query a number
        shared_future<int> f = async(queryNumber);

        // start three threads each processing this number in a loop
```

```
        auto f1 = async(launch::async,doSomething,'.',f);
        auto f2 = async(launch::async,doSomething,'+',f);
        auto f3 = async(launch::async,doSomething,'*',f);

        // wait for all loops to be finished
        f1.get();
        f2.get();
        f3.get();
    }
    catch (const exception& e) {
        cout << "\nEXCEPTION: " << e.what() << endl;
    }
    cout << "\ndone" << endl;
}
```