```
//
**********************************************************************
**** //
//
//
//                tests.cpp for GlobalBanksters
United                        //
//                Created on  : Thu Nov 20 23:45:02
1989                      //
//                Last update : Wed Jan 04 09:23:52
1992                      //
//                Made by : Brad "Buddy" McLane
<bm@gbu.com>                  //
//
//
//
**********************************************************************
**** //

#include <vector>
#include <algorithm>
#include <functional>
#include "Account.hpp"


int              main( void ) {

        typedef std::vector<Account::t>
                           accounts_t;
        typedef std::vector<int>
                           ints_t;
        typedef std::pair<accounts_t::iterator, ints_t::iterator>
acc_int_t;

        int      const                             amounts[]        =
{ 42, 54, 957, 432, 1234, 0, 754, 16576 };
        size_t const
amounts_size( sizeof(amounts) / sizeof(int) );
        accounts_t                                 accounts( amounts,
amounts + amounts_size );
        accounts_t::iterator     acc_begin         = accounts.begin();
        accounts_t::iterator     acc_end           = accounts.end();

        int      const                    d[]                       =
{ 5, 765, 564, 2, 87, 23, 9, 20 };
        size_t const             d_size( sizeof(d) / sizeof(int) );
        ints_t                            deposits( d, d + d_size );
        ints_t::iterator dep_begin        = deposits.begin();
        ints_t::iterator dep_end          = deposits.end();
        int      const                    w[]                        =
{ 321, 34, 657, 4, 76, 275, 657, 7654 };
        size_t const             w_size( sizeof(w) / sizeof(int) );
        ints_t                            withdrawals( w, w +
w_size );
        ints_t::iterator wit_begin        = withdrawals.begin();
        ints_t::iterator wit_end          = withdrawals.end();

        Account::displayAccountsInfos();
        std::for_each( acc_begin, acc_end,
std::mem_fun_ref( &Account::displayStatus ) );

        for ( acc_int_t it( acc_begin, dep_begin );
                   it.first != acc_end && it.second != dep_end;
                   ++(it.first), ++(it.second) ) {

                 (*(it.first)).makeDeposit( *(it.second) );
        }

        Account::displayAccountsInfos();
        std::for_each( acc_begin, acc_end,
std::mem_fun_ref( &Account::displayStatus ) );

        for ( acc_int_t it( acc_begin, wit_begin );
                   it.first != acc_end && it.second != wit_end;
                   ++(it.first), ++(it.second) ) {

                 (*(it.first)).makeWithdrawal( *(it.second) );
        }

        Account::displayAccountsInfos();
        std::for_each( acc_begin, acc_end,
std::mem_fun_ref( &Account::displayStatus ) );

        return 0;
}


//
**********************************************************************
**** //
// vim: set ts=4 sw=4 tw=80
noexpandtab:                                        //
// -*- indent-tabs-mode:t;
-*-
// -*- mode: c++-mode;
-*-
// -*- fill-column: 75; comment-column: 75;
-*-
//
```

```
************************************************************************
**** //
```