3460:209-010 Fall 2011 Lab 7 Report

Name:
UANET id: ig11

Generated: Fri Nov 13 09:07:09 EST 2015

Success!

Note: First column is revision number when that line was last changed.

```
------------ ig11/Labs/Lab7/algorithm.hpp ------------
#ifndef ALGORITHM_HPP
#define ALGORITHM_HPP
#include <iostream>

using namespace std;

template<typename input, typename T>

input find (input first, input last, const T& val) {
        while(first != last) {
                if(*first == val) return first;
                ++first;
        }

        return last;
}

template<typename test>
bool equals(test first1, test last1, test first2, test last2) {
        int i = 0;
        int j = 0;
        while(first1!=last1) {
                if(*first1 != *first2){
                        return false;
                }
        ++first1;
        ++first2;
        }
        return true;
}

template<typename T>
const T& minimum(const T& first, const T& second) {
        if(first < second)
                return first;
        else
                return second;
}

template<typename T>
T min_element(T first, T last) {
        T min = first;
        ++first;
        while(first != last) {
                if(!(*min < *first))
                        min = first;
        ++first;
        }
        return min;
}

template<typename T>
T copies(T first, T last, T d_first) {
        while(first != last) {
                *d_first = *first;
                ++first;
                ++d_first;
        }

//      cout << "first : " << *(--first) << " d_first : " << *(--d_first) << "\n";
        return  (d_first);
}

template<class InputIt, class T>
T accumulate(InputIt first, InputIt last, T init)
{
    for (; first != last; ++first) {
        init = init + *first;
    }
    return init;
}

#endif

------------ ig11/Labs/Lab7/algorithm.cpp ------------
#include "algorithm.hpp"

------------ ig11/Labs/Lab7/CMakeLists.txt ------------
project(Lab7 CXX)

cmake_minimum_required(VERSION 2.8)

set(CMAKE_CXX_FLAGS "-std=c++11")


add_executable(lab7 algorithm.cpp main.cpp)

------------ ig11/Labs/Lab7/main.cpp ------------
#include <iostream>
#include <vector>
#include <cassert>
#include <string>

#include "algorithm.hpp"

void test_find()
{
    // Range of strings
    std::vector<std::string> str_1={"Hi", "this", "is", "a", "test", "for", "equal"
};

    // True Case
    std::string s= "Hi";
    auto StrIterator = find(str_1.begin(),str_1.end(),s);
    assert(StrIterator != str_1.end());
    assert(s == *StrIterator);

    // False Case
    s="None";
    StrIterator = find(str_1.begin(),str_1.end(),s);
    assert(StrIterator == str_1.end());

    // Range of ints
    std::vector<int> val_1 = {1,2,3,4,5,6,7,8,9,10};
    // True Case
    int val=5;
    auto IntIterator=find(val_1.begin(),val_1.end(),val);
    assert(IntIterator != val_1.end());
    assert(*IntIterator == val);

    // False case
    val=12;
    IntIterator=find(val_1.begin(),val_1.end(),val);
    assert(IntIterator == val_1.end());
}
```

3460:209-010 Fall 2011 Lab 7 Report

```
void test_equals()
{
    // Test for range of strings
    std::vector<std::string> str_1={"Hi", "this", "is", "a", "test", "for", "equal"
};
    std::vector<std::string> str_2={"Hi", "this", "is", "a", "test", "for", "equal"
};
    std::vector<std::string> str_3={"Hi", "test", "for", "equal"};
    std::vector<std::string> str_4={"Hi", "this", "is", "a"};
    assert(equals(str_1.begin(),str_1.end(),str_2.begin(),str_2.end()));
    assert(!equals(str_3.begin(),str_3.end(),str_4.begin(),str_4.end()));

    // Test for range of integers
    std::vector<int> val_1 {1,2,3,4,5,6,7,8,9,10};
    std::vector<int> val_2 {1,2,3,4,5,6,7,8,9,10};
    std::vector<int> val_3 {1,2,3,4,5,6};
    std::vector<int> val_4 {5,6,7,8,9,10};
    assert(equals(val_1.begin(),val_1.end(),val_2.begin(),val_2.end()));
    assert(!equals(val_3.begin(),val_3.end(),val_4.begin(),val_4.end()));
}

void test_min()
{
    int a=10,b=20;
    assert(minimum(a, b) ==a);

    std::string s1="foo",s2="bar";
    assert(minimum(s1, s2) == s2);
}
void test_min_element()
{
    std::vector<int> val_1 = {12,2,3,4,5,1,6,7,8,9,10};
    auto IntIterator = min_element(val_1.begin(),val_1.end());
    assert(*IntIterator == 1);

    std::vector<std::string> str_1={"hi", "this", "is", "a", "test", "for", "equal"
};
    auto StrIterator = min_element(str_1.begin(),str_1.end());
    assert(*StrIterator == "a");
}
void test_copies()
{
    // Test for range of integers
    std::vector<int> val_1 = {12,2,3,4,5,1,6,7,8,9,10};
    std::vector<int> val_2(11);
    auto it = copies(val_1.begin(),val_1.end(),val_2.begin());
    assert(*(val_1.begin())==*(val_2.begin()));
    assert(it == val_2.end());

    // Test for range of Strings
    std::vector<std::string> str_1={"Hi", "this", "is", "a", "test", "for", "equal"
};
    std::vector<std::string> str_2(7);
    auto StrIterator = copies(str_1.begin(),str_1.end(),str_2.begin());
}
int main()
{
    test_find();
    test_equals();
    test_min();
    test_min_element();
    test_copies();
    std::cout << "Success!" << std::endl;
    return 0;
}
------------------------------------------------------------------------
r1365 | ig11 | 2015-11-10 07:51:52 -0500 (Tue, 10 Nov 2015) | 1 line

reverted to given main
------------------------------------------------------------------------
r1364 | ig11 | 2015-11-10 07:51:32 -0500 (Tue, 10 Nov 2015) | 1 line

fixed all bugs
------------------------------------------------------------------------
r1324 | ig11 | 2015-11-08 23:30:33 -0500 (Sun, 08 Nov 2015) | 1 line

commit
------------------------------------------------------------------------
r1323 | ig11 | 2015-11-08 23:30:04 -0500 (Sun, 08 Nov 2015) | 3 lines

test ommit


------------------------------------------------------------------------
r1206 | hvn1 | 2015-11-02 15:11:36 -0500 (Mon, 02 Nov 2015) | 1 line

lab 7 files for both sections
------------------------------------------------------------------------
```