3460:209-010 Fall 2011 Lab 9 Report

Name:
UANET id: ig11

Generated: Thu Dec 3 17:35:03 EST 2015

Success!

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

```
------------ ig11/Labs/Lab9/algorithm.hpp ------------
#ifndef ALGORITHM_HPP
#define ALGORITHM_HPP
#include <iostream>
#include <vector>
#include <string>
using namespace std;
template<typename T, typename input>
T find(T first,const T end, const input& val) {
        if(first == end){
                return first;
        } else if(*first == val) return first;
        else return find(++first, end, val);


}



template<typename T>
bool equals(T first1, T last1, T first2, T last2) {
        if(first1 != last1){
                if(*first1 != *first2){
                        return false;
                } else{
                        equals(++first1, last1, ++first2, last2);
                }
        }
        else return true;

}


template<typename T>
T min_element(T first, T end) {
        if(first == end - 1) return first;
        T min = min_element(++first, end);
return (*first < *min) ? first : min;

}

template<typename T>
T copies(T first, T last, T d_first) {
        if(first == (last -1)) {
                *d_first = *first;
                return d_first + 1;
        } else {
                *d_first = *first;
                return copies(++first, last, ++d_first);
        }
}

#endif

------------ ig11/Labs/Lab9/algorithm.cpp ------------
#include "algorithm.hpp"

------------ ig11/Labs/Lab9/CMakeLists.txt ------------
project(Lab9 CXX)

cmake_minimum_required(VERSION 2.8)

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


add_executable(lab9 algorithm.cpp main.cpp)

------------ ig11/Labs/Lab9/main.cpp ------------
#include <iostream>
#include <vector>
#include <cassert>
#include <string>
#include "algorithm.cpp"

using namespace std;
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());
}


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_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_element();
        test_copies();
        std::cout << "Success!" << std::endl;
        return 0;
}
------------------------------------------------------------------------
r1915 | ig11 | 2015-12-01 15:54:09 -0500 (Tue, 01 Dec 2015) | 1 line

LAST COMMIT
------------------------------------------------------------------------
r1883 | ig11 | 2015-11-30 13:59:23 -0500 (Mon, 30 Nov 2015) | 1 line


------------------------------------------------------------------------
r1882 | ig11 | 2015-11-30 13:59:15 -0500 (Mon, 30 Nov 2015) | 1 line


------------------------------------------------------------------------
r1850 | ig11 | 2015-11-26 22:05:15 -0500 (Thu, 26 Nov 2015) | 1 line


------------------------------------------------------------------------
r1537 | hvn1 | 2015-11-16 15:17:37 -0500 (Mon, 16 Nov 2015) | 1 line

lab 9 folders and files for both sections
------------------------------------------------------------------------
```