```cpp
//
//  main.cpp
//
//  Michael Campbell
//  7/31/11
//

#include <iostream>
#include <cstdlib>
#include <vector>
#include <list>
#include "date.h"
#include "sorter.h"
using namespace std;
//using namespace Dater;

template <class ITERATOR>
void print(ITERATOR first, ITERATOR last)
{
    for (; first != last; ++first) {
        cout << &first << "\n";
    }
}

int main()
{
    int a1[] = {10, 30, 20, 50, 40};
    const size_t n1 = sizeof a1 / sizeof a1[0];
    sorter(a1, a1 + n1);
    print(a1, a1 + n1);
    cout << "\n";

    const date a2[] = {
        date(October,   29, 1929),
        date(July,      20, 1969),
        date(July,       4, 1776),
        date(September, 11, 2001),
        date(December,   7, 1941)
    };
    const size_t n2 = sizeof a2 / sizeof a2[0];

    vector<date> v(a2, a2 + n2);
//    sorter(v.begin(), v.end());
    print(v.begin(), v.end());
    cout << "\n";

    list<date> li(a2, a2 + n2);
    //sorter(li.begin(), li.end());   //won't compile
    li.sort();
    print(li.begin(), li.end());
    return EXIT_SUCCESS;
}
```

```
//
//  date.h
//
//  Michael Campbell
//  7/31/11
//

#ifndef DATEH
#define DATEH
#include <iostream>
using namespace std;

//namespace Dater {

    enum e_month {
        January = 1,
        February = 2,
        March = 3,
        April = 4,
        May = 5,
        June = 6,
        July = 7,
        August = 8,
        September = 9,
        October = 10,
        November = 11,
        December = 12
    };

    class date {
        int year;
        e_month month;              //1 to 12 inclusive
        int day;                //1 to date_length[month] inclusive

    public:
        date(e_month initial_month, int initial_day, int initial_year);
        void next(int count);    //Go count days forward.
        void next();             //Go one day forward.
        void print() const {cout << month << "/" << day << "/" << year;}
        void setdate(e_month e_month_name, int some_day, int some_year);

        friend ostream& operator <<(ostream& outputStream, const date&
            someDate);
    };
//}
#endif
```

```
//
//  date.cpp
//
//  Michael Campbell
//  7/31/11
//

#include <iostream>
#include <cstdlib>
#include "date.h"
using namespace std;
//using namespace Dater;

const int date_length[] = {
    0,   //dummy element so that January will have subscript 1
    31,   //January
    28,   //February
    31,   //March
    30,   //April
    31,   //May
    30,   //June
    31,   //July
    31,   //August
    30,   //September
    31,   //October
    30,   //November
    31    //December
};

void date::setdate(e_month e_month_name, int some_day, int some_year)
{
    day = some_day;
    year = some_year;
    month = e_month_name;
}
date::date(e_month initial_month, int initial_day, int initial_year)
{
    if (initial_month < 1 || initial_month > 12) {
        cerr << "bad month " << initial_month << "/" << initial_day
        << "/" << initial_year << "\n";
        exit(EXIT_FAILURE);
    }

    if (initial_day < 1 || initial_day > date_length[initial_month]) {
        cerr << "bad day " << initial_month << "/" << initial_day
        << "/" << initial_year << "\n";
        exit(EXIT_FAILURE);
    }

    year = initial_year;
    month = initial_month;
    day = initial_day;
}

void date::next(int count)
{
```

```
    //Call the other next count times.
    while (--count >= 0) {
        next();
    }
}

void date::next()
{
    //Move to the next date.
    if (++day > date_length[month]) {
        day = 1;
        if (++month > 12) {
            month = January;
            ++year;
        }
    }
}
//ostream& operator <<(ostream& outputStream, const Complex& complex_num){
//    //
//    cout << complex_num.real;
//    if (complex_num.imaginary < 0) {
//        cout << " - ";
//    }else if (complex_num.imaginary >=0){ cout << " + ";
//    }
//    cout << complex_num.imaginary << "i" << endl;
//    return outputStream;

ostream& operator <<(ostream& outputStream, const date& someDate){

    cout << someDate.month << "/" << someDate.day << "/" << someDate.year;


    return outputStream;

}
```

```
//
//  Sorter.h
//
//  Michael Campbell
//  7/31/11
//

#ifndef SORTERH
#define SORTERH
#include <iterator>   //for iterator_traits
using namespace std;

template <class ITERATOR>
    void sorter(ITERATOR first, ITERATOR last)
    {
        while (first != --last) {                               //change me
            for (ITERATOR it = first; it < last; ++it) {    //change me
                if (it[1] < *it) {
                    const typename iterator_traits<ITERATOR>::value_type
                    temp = *it;
                    *it = it[1];
                    it[1] = temp;
                }
            }
        }
    }

#endif
```