

Hack 13.0

Computer Science I

Searching & Sorting

Department of Computer Science & Engineering
University of Nebraska–Lincoln

Introduction

Hack session activities are small weekly programming assignments intended to get you started on full programming assignments. Collaboration is allowed and, in fact, *highly encouraged*. You may start on the activity before your hack session, but during the hack session you must either be actively working on this activity or *helping others* work on the activity. You are graded using the same rubric as assignments so documentation, style, design and correctness are all important. This activity is **due at 23:59:59 on the Friday** in the week in which it is assigned according to the CSE system clock.

Problem Statement

You will use the implementation of your Airport data model from a previous hack to develop several reports that will require you to sort and search (a subset of) the International Civil Aviation Organization database for particular airports.

We have provided an updated header file, `airport.h` with the specific functions that you need to implement. First, you'll need to implement 8 different comparator functions for your `Airport` structure. The details of the expected orders are documented in the header file.

Second, you'll need to write code that produces several reports. All your code needs to be placed into the following function:

```
void generateReports(Airport *airports, int n);
```

which takes an array of `Airport` structures and produces the following reports which should all be output to the standard output.

- To help you troubleshoot, you should print out all the structures in the original order.

- Sort the airports by each of the 8 comparators and print them out (8 reports total)
- Search for and print out the airport in the array that is closest (via air distance) to Lincoln. Lincoln is located at 40.8507N, 96.7581W.
- Search for and print out the airport that is the geographic west-east median of the given airports with respect to longitude.
- Search for an airport located in the city New York and print it out if it exists. If no such airport exists, print out an appropriate message.
- Search for an airport located in Canada (country code “CA”) and print it out if it exists. If no such airport exists, print out an appropriate message.
- Search for an airport whose type is `large_airport` and print it out if it exists. If no such airport exists, print out an appropriate message.

Instructions

- Place all of your function definitions in a source file named `airport.c` and hand it in with your header file, `airport.h`. You may add any utility functions you wish but you must *not* change any of the signatures of the required functions.
- In addition, you must create a main test driver program that demonstrates your reports with at least 5 airports. Name this file `airportReport.c` and hand it in.
- You are encouraged to collaborate any number of students before, during, and after your scheduled hack session.
- You may (in fact are encouraged) to define any additional “helper” functions that may help you.
- Include the name(s) of everyone who worked together on this activity in your source file’s header.
- Turn in all of your files via webhandin, making sure that it runs and executes correctly in the webgrader. Each individual student will need to hand in their own copy and will receive their own individual grade.