CSC 112 Lab 5
Inventory Database v1.0
Fall 2007

Due
3:00pm Friday, 10/18

1 Overview

A company maintains a database of products kept in a warehouse. Each product has a unique ID number and its presence in the database indicates the company stocks the item. You must write a program that maintains the database. This will include removing duplicate entries, sorting, and adding or deleting ID numbers from the database.

1.1 Database Structure

The product database is a simple ASCII file. The file called `invent.dat` contains the list of ID numbers, one number (integer) per line. ID numbers in the file are in random order and each number may appear multiple times in the file (you will remove these duplicates). A second file called `update.dat` contains a list of *action-ID* pairs. Each line of the file stores a letter (action-code) and an ID number. There are two possible action-codes, *add* and *delete*. `'A'` followed by an ID number indicates the ID should be added to the database. If the ID number is already in the database, then the number is not added. `'D'` followed by an ID number indicates the ID should be deleted from the database. If the ID number is not in the database, then the action is ignored. Uppercase or lowercase action codes are allowed. Actions occur in sequential order as placed in the file.

`invent.dat`

```
102020
220300
102020
222944
33449
```

`update.dat`

```
D 102020
a 567080
A 102020
x 222944
a 102020
d 33449
```

`final.dat`

```
102020
220300
222944
567080
```

2 Program Description

You must write a C++ program that reads the database file `invent.dat` and stores the **unique** ID values in ascending order in an array. Assume no more than 2000 unique ID values exist; therefore, size your arrays using this value (using `#define`). Once `invent.dat` has been read the program must indicate the number of ID numbers stored, then update the information in the array using the file `update.dat` After the database (array) is updated, your program should display the number of items stored. Write the updated information in another file called `final.dat` Therefore, the file will contain a list of unique ID numbers sorted in ascending order (after the appropriate add and deletes are performed).

Terminal

```
Database has 4 ID numbers
--------------------------------------------------
Updated database
Database has 4 ID numbers
```

3 Programming Points

You **must** adhere to all of the following points to receive credit for this program.

1. Turn-in (print-outs and electronically) the files for this program.
2. The assignment will consist of 4 files.
   - `main.cpp` contains the main function.
   - `database.h` contains the database function prototypes.
   - `database.cpp` contains the database function definitions.
   - `makefile` a makefile that compiles the program and generates the executable called `lab5`
3. The main function can only contain variable declarations, function calls, selection structure, and a repetition structure.
4. **All arrays must be statically sized for 2000 elements**.
5. Must adhere to documentation style and standards.