Programming Assignment
2003 Comprehensive Exam

Department of Computer Science
Brown University
Providence, RI 02912

Out: **Monday, January 13, 2003 at 9:00 am**
Due: **Thursday, January 16, 2003 at 9:00 pm**
Presentations: **Friday, January 17, 2003 and**
Tuesday, January 21, 2003

1 Task

The purpose of this assignment is to implement an application that supports searching for citations in a collection of BibTeX bibliographies. BibTeX is a program and the corresponding file format designed for handling bibliographic citations in the LaTeX document preparation system. The format is character based, so it can be used by any program. It is field based and new fields can be added (the BibTeX program will ignore unknown fields), so it is expandable. It is probably the most common format for bibliographies on the Internet.

A BibTeX bibliography is a text file with extension `.bib` containing a collection of BibTeX *entries* and *abbreviations*. An example of a BibTeX entry is as follows:

```
@article{lp-lppsi-77
, author =      "D. T. Lee and F. P. Preparata"
, title =       "Location of a point in a planar
                 subdivision and its applications"
, journal =     "SIAM J. Comput."
, volume =      6
, number =      3
, year =        1977
, pages =       "594--606"
}
```

In this example, `@article` denotes the *type*, `lp-lppsi-77` is the *key*, and `author`, `title`, `journal`, `volume`, `number`, `year`, and `pages` are the names of the *fields* of the entry.

Each field in a BibTeX entry has a name and an associated value, which is enclosed either by braces or by double quotes. If a field value consists entirely of numerals, then the braces or quotes can be omitted. The entire entry, except for the type, is enclosed by braces or parentheses. They key and the fields are separated by commas. The field name is separated from its value by an equal sign. The case of letters in the type, key and field names is ignored. A space may appear before and/or after the key and the fields. Tabs and end-of-line characters are equivalent to spaces and multiple spaces are equivalent to a single space. Thus, the sample BibTeX entry above can also be written as follows:

```
@ARTICLE(LP-lppsi-77, Author = "D. T. Lee and F. P. Preparata",
    Title = {Location of a point in a planar subdivision and its
    applications}, journal = "SIAM J. Comput.", volume = {6},
    number = "3", year = 1977, pages = "594--606")
```

The author field contains a list of author names separated by the word `and`. If a name has a comma, then the last name consists of the word(s) before the first comma, else, the last name is the last word plus all the lowercase words immediately preceding it. For example, the following author fields are equivalent:

```
author = "James D. Foley and Andries van Dam
            and Steven K. Feiner and John F. Hughes"

author = { Foley, James D. and van Dam, Andries
            and Steven K. Feiner and John F. Hughes }
```

A sequence of words between braces is treated as a single word. Thus, Pascal Van Hentenryck can be written in BibTeX either as `Pascal {Van Hentenryck}` or `Van Hentenryck, Pascal`. Writing `Pascal Van Hentenryck` in the author field of a BibTeX entry is incorrect since the BibTeX program will think that Van is Pascal's middle name.

An example of a BibTeX abbreviation is shown below:

```
@string{ jgaa = "Journal of Graph Algorithms and Applications" }
```

The above abbreviation makes the following field definitions equivalent:

```
journal = "Journal of Graph Algorithms and Applications"

journal = jgaa
```

An abbreviation can be used to replace an entire field value. It should not be enclosed in braces or double quotation marks. Entries and abbreviations can be arbitrarily interleaved in a BibTeX file. However, it is customary to put abbreviations before entries.

Details on the BibTeX format are given in Appendix B (pages 155-164) of the LaTeX reference book (*LaTeX: A Document Preparation System*, 2nd Ed., by Leslie Lamport, Addison-Wesley, 1994). See also `http://www.ecst.csuchico.edu/~jacobsd/bib/formats/bibtex.html`.

You can find various BibTeX bibliographies in the directory `/home/rt/lib/bibtex` and on the Web at `http://liinwww.ira.uka.de/bibliography/`.

2 Functionality

Your application should maintain a collection of BibTeX bibliographies and support the following *search, insertion* and *deletion* operations.

Search. This operation finds and returns all the BibTeX entries in the collection that satisfy the search criteria and, for each citation, the bibliography containing it. The following *simple searches* should be supported:

- *By author*: finds and returns all the citations with a given author's last name.
- *By title*: finds and returns all the citations with a given word in the title.
- *By year*: finds and returns all the citations with a given year.

The results of a search should be displayed sorted according to the user's preferences. For this purpose, you should provide a mechanism for selecting one of the following *sort fields*:

- *Last name of first author.*
- *Title.*
- *Year.*

Furthermore, *composite searches* specified by the *conjunction* (i.e., AND) or *disjunction* (i.e., OR) of two simple searches should also be supported.

You should match entire words in the search operation. For instance, the keyword "*net*" should match the title "*Distributed Net Applications Create Virtual Supercomputers*" but not the title "*Pervasive Networking*". In addition, your searches should be *case insensitive*.

A *search history* (i.e., the list of previously performed searches) should be maintained so that previous searches can be easily repeated. Note that you do not need to cache the results of previous searches.

Insertion. This operation takes a BibTeX bibliography as input and adds it to the current collection. You are free to handle equivalent entries in whichever way you want. For example, you can decide not to detect whether two entries are equivalent and allow multiple equivalent entries.

Deletion. This operation removes a given BibTeX bibliography from the current collection (i.e., you should remove all the BibTeX entries that belong to that collection).

You are allowed to make the following assumptions and simplifications:

- You can assume that the bibliography specified in an insertion operation has the correct format.
- You do not have to completely parse the BibTeX bibliography to be inserted. You only have to identify the entries and, for each entry, the last names of the authors, the words in the title and the year. You do not have to identify other fields.
- You can come up with your own reasonable definition of what exactly constitutes a "word" of a title. This definition should allow useful searches without making the parsing of titles too cumbersome.
- You do not have to process or handle in a special way LaTeX formatting commands within field values. For example, you do not have to convert `M{\"u}ller` to `Mueller`. Also, a search by title for the word `Bezier` does not have to retrieve an entry with the following title field:

  ```
  title = "Data structures to support {B{\'e}zier}-based modeling"
  ```

- Abbreviations can be ignored. That is, you can ignore author, title and year fields whose values are abbreviations.
- If the value of a year field is not a numeral (e.g., an entry may have "to appear" as the year), you can ignore it. Alternatively, you can allow searching for arbitrary words in the year field.
- You do not have to implement advanced information retrieval techniques such as stemming (identifying words that differ only by common suffixes, e.g., technology and technological) and ignoring stopwords (words of little semantic content, such as articles and prepositions, that occur too frequently to be useful in searching). Thus, you can treat "network" and "networks" as distinct words and you can allow searches for the word "of".

3 User Interface

The user interface should support the operations described in Section 2:

- *Search:* the user specifies the search parameters (including the sort field).

- *Insertion*: the user provides the path in the file system (or the URL) of the bibliography to be added. Feedback should be provided if the bibliography specified does not exist.
- *Deletion*: the user selects the bibliography to be deleted from the list of existing bibliographies in the collection.

If you implement a standalone application, you can provide either a graphical user interface or a textual command-line interface.

It is important that your interface be *usable*, which is a quality that goes much beyond the visual attractiveness of your interface. In particular, your interface should exhibit, as much as possible, the following attributes. First, it should be easily learnable in that it should not take long for a new user to become productive with the interface. Second, it should be robust; i.e., it should be tolerant of user error. Finally, the interface should easily recover from user errors.

4 Documentation

You code should be well commented. This does not mean that you should insert a comment per each line of your code. Rather, you should use your comments to ease the understanding of the high-level functionality of your components as well as the relatively complicated pieces. In addition to commenting your code, you should prepare the following documents for submission:

- *User's Guide:* This document should first provide a functional description of your application. It should then provide instructions for "normal" usage; it should describe how to set the application up and how end-users would use the application. It should also describe error conditions/messages and how to recover from them (if at all possible).
- *Programmer's Guide*: This guide should contain the basic design and structure of your implementation. In particular, you should discuss the components of your system and how they interact (i.e., how they share data and how they interface with each other). All the external libraries and tools used (see Section 6) should be clearly identified. You should discuss the primary data structures and the algorithms employed and their asymptotic running times. The testing strategy adopted should be described. Sufficient details should be provided so that a programmer unfamiliar with your code can easily understand and extend your code. You should also discuss the limitations of your code (e.g., additional assumptions about the input format, errors you do not handle, etc.).

Both documents should be typeset. The User's Guide should be approximately 2-3 pages, and the Programmer's Guide should be approximately 5-6 pages. However, you should take these numbers merely as guidelines and not as strict requirements—you are welcome to use fewer or more pages, as you deem necessary.

5 Design and Implementation

The application can be realized either as a standalone program or as a web-based application. In either case, the collection of BibTeX bibliographies should be stored in a format that supports fast searches. You can use either internal-memory data structures (e.g., a hash table or a balanced search tree) or external-memory storage (e.g., a relational database).

For Web-based applications, you should ensure that the search structure persists across successive HTTP requests. Some platforms (e.g., Java servlets and the PLT Scheme Web server) provide a transparent mechanism for data persistency. Alternatively, you can explicitly manage external-memory storage.

6 Mechanics

You may write your application in any programming language (or combination of programming languages). You can use any libraries that are considered part of that language (for example, the STL is considered part of C++; all classes in java.* or javax.* are considered part of Java). Furthermore, you can use any *generic* software tools, packages, and systems (such as lexical analyzers, parsers, libraries of data structures and algorithms, Perl modules, DBMSs, Web servers, JSP/ASP, CGI, etc.). You are also allowed to download and install programming languages and support software and tools if they are not readily available on our system.

You cannot use *specific* software tools designed to solve the problem or replace the application that you are asked to implement. For example, you cannot simply extend or build an interface to the biblook system (see `http://sourceforge.net/projects/biblook/` and the command `/usr/bin/biblook`). While you may inspire your design to biblook, no endorsement is made here about the suitability of the design of the biblook system for your assignment. Thus, if you decide to inspire your design to biblook, you should be prepared to fully explain and defend your choices.

For developing your application, you may use any machines that you normally have access to at Brown or you may use your own personal machine. However, your application should run on a standard departmental Maxbuilt system, under either Linux or Windows, from the submission directory (see Section 7).

You should test your application at least on all the bibliographies in the directory `/home/rt/lib/bibtex`. You may want to create and/or download (e.g., from `http://liinwww.ira.uka.de/bibliography/`) additional test bibliographies. Consider using the biblook system as an aid for testing your application.

Except for the allowed generic libraries and tools discussed above, all work that you hand in must be your own. You should not discuss your work or ideas with anyone else. You should not share code or ask others for advice on your design or code.

If you have questions, you should contact a member of the Programming Exam Committee. Corrections, clarifications and answers to questions of general interest will be posted to the comps news group (`brown.cs.comps`).

7 Submission

You should submit the following materials by **9:00pm EST on Thursday, January 16, 2003**:

1. **Hardcopy documents**:
    - Printout of the User's Guide
    - Printout of the Programmer's Guide

    The hardcopy documents should be given in person to Fran Palazzo (before 4:30pm) or to Roberto Tamassia (between 7pm and 9pm):

2. **Project files**:
    - Source files
    - Executables
    - Test data
    - Documentation

The project files should be placed in the **submission directory**

```
/pro/comps/name
```

where `name` is your login name. Please do not modify the permissions of this directory, which should remain accessible only to you. The submission directories will be locked at 9:00pm EST on Thursday, January 16 and made available to the Programming Exam Committee. You will be graded on what is there at the time.

8 Presentation

You will be asked to give a presentation of your application to the Programming Exam Committee. The format of the presentation will be informal: you will explain your design choices, conduct a walkthrough of your application by running it from the submission directory, outline your testing strategy, and answer questions from the committee members.

You should not prepare slides or bring any other material besides what you have already submitted on January 16.

The committee will set up the schedule of the presentations, which will be held on the following days: **Friday, January 17**, and **Tuesday, January 21, 2003**.

9 Grading

Your grade will be approximately computed as follows:

40%	Functionality
25%	Efficiency
15%	Documentation
10%	Design
10%	Usability

You will be graded relative to the programming language(s) and tools you use in the implementation.

Appendix: Resources

You may find the following software resources and tools useful for your assignment:

- BibTeX tools (includes format verifiers and parsers):
  http://www.ecst.csuchico.edu/~jacobsd/bib/tools/bibtex.html
- Biblook:
  http://sourceforge.net/projects/biblook/
- Lexer and Parser generators (includes Lex and Yacc):
  http://catalog.compilertools.net/lexparse.html
- JavaCC: The Java Parser Generator:
  http://lml.ls.fi.upm.es/manuales/javaccdocs/
- The Common Gateway Interface (CGI):
  http://hoohoo.ncsa.uiuc.edu/cgi/overview.html
- Java Server Pages (JSP):
  http://java.sun.com/products/jsp/
- MS Active Server Pages (ASP):
  http://www.asp.net/
- MySQL (open source database):
  http://www.mysql.com/
- Berkeley DB (open source database):
  http://www.sleepycat.com/

Good luck and have fun!