CSI 2131 File Management Winter 2006
Computer Science University of Ottawa

Homework Assignment #3 (100 points, weight 10%)

due: Friday, April 7 at midnight (via webCT)

You will modify the primary and secondary index from the solution for assignment#2-program 2, substituting simple indexes by hashed indexes.
Please, refer to assignment#2, for the full specification of the given files and indexes.
You must use the solution provided as a startup code.

1 Written part: hashing functions (20 marks)

The written part will consist of the following tasks summarized on a report:

- Design a hashing function h_1 for the primary key (student number). The hashing function h_1 should use both the key (student number) and N_1, the number of hashing addresses (N_1 is arbitrary).
- Design a hashing function h_2 for the secondary key (last name). The hashing function h_2 should use both the key (last name) and N_2, the number of hashing addresses (N_2 is arbitrary).
- Test your program (described in section 2) for various values of N_1 and N_2, in order to determine the best value for each of these parameters. The results should be included in your report. Be prepared to change your hashing functions if your first choice doesn't work well for the given keys.

Your report (word or textfile not exceeding 2 pages) shound contain, for each hash function:

- The hash function description.
- A table for each hash function h_i which contains for each tested hashtable size N_i, the average search length calculated by your program. Based on these results, you should recommend and justify a good choice for the value of N_1 and N_2.

Marks will be given for quality of the solution and of the analysis.

2 Program: Hashed primary and secondary indexes (80 marks)

Like in assignment#2, program 2, your program will first build the primary and secondary index files, through a sequence of insertions to each index file. Then, your program will process a sequence of searches which can be by primary or by secondary key.

Before you process the searches, calculate and output the average search length for each of the hashed indexes, as defined in class. You could store in each index class some information

that is updated at time of insertions which allows you to easily obtain the average search length when it is time to print it.
You will change the primary index file to be a hashed index file (no buckets) using hashing function h_1 with N_1 addresses, and with collisions handled by progressive overflow.
You will change the secondary index file to have its **Secondary Key Index File** as a hashed index file (with bucket size 2) using hashing function h_2 with N_2 addresses (space for $2 \times N_2$ hashed records), and with collisions handled by progressive overflow. The **Lists file** will not change, so the secondary index is technically organized as inverted lists, with the main index being hashed rather than a simple index.
Remember that before inserting keys into a hashed file, the hashed file with the specified size has to be created.

Program input
You will prompt the user for datafile name, search file name, primary index file name, primary index hashtable size (N_1), secondary key index file name, secondary index hashtable size (N_2), and secondary key lists file name.

2.1 Submission instructions

By submitting this assignment, you are automatically acknowledging understanding of the policy on plagiarism available from the course web page.
EVERY FILE MUST CONTAIN A HEADER WITH Student Name, Student Number, and Course. Guidelines on style and documentation should be followed, as in assignment#1 and #2.
Create a directory/folder named `a3` containing the source files for your program (include all .cpp and .h files that you create) and the file for your report. Zip the folder `a3` and submit the zipped file as your assignment#3 submission to webCT (only one file is submitted).