CIS*2520 Data Structures Assignment 4

Due: Sunday, Nov. 17, 2019, 11:59pm
Version 1.03 (changes highlighted in yellow)

For this assignment you will be building hash functions and evaluating their performance.

THE DATA
You will use data from https://data.delaware.gov/Licenses-and-Certifications/Professional-and-Occupational-Licensing/pjnv-eaih which was used in the in-class example hash.c. This file has been uploaded to the linux server in directory /home/courses/cis2520 .

THE HASH TABLES
Build 3 different hash functions. You will be graded in part, on how efficient your hash functions are (the fewer the collisions while you are building the hash, the better).

In a file called, a4.c you will write three hash functions, call them hash1, hash2, and hash3. Each function will take a string and a hash_size as it's arguments, and return an integer between zero and one less than the hash_size. Provide a header file for the hash functions called a4.h:

e.g. int hash1(char *string, int hash_size)

DEVELOPMENT
Modify the hash.c example code to create a table with the following columns:

(1) last_name,
(2) first_name,
(3) license_no,
(4) license_type, and
(5) issue_date.

Design a hash function to find the last_name, license_no, and issue_date (aka keys).

Count the number of collisions that occur as you build the hash table using linear probing when the hash has its first collision. Try to develop one function that performs well for each key. In particular, try to develop one function for each lookup key that works better than the str2int function in hash.c.

ORGANIZING YOUR CODE
Create a4.c and a4.h files that define the 3 hash functions hash1, hash2 and hash3. These files should compile to a a4.o file without warnings or errors when compiled with the statement:

gcc -Wall -ansi -pedantic -c a4.c -o a4.o
Do not include char2int, str2int, read_records, build_hash, or main functions in your files. Do not include structures in your code.

SUBMITTING YOUR CODE
Submit your .c, and .h files via git.

Put your full name, student ID number, and uoguelph e-mail at the top of each file.

Use consistent indenting, formatting, commenting, naming and other good programming practises to make your code readable.

IMPORTANT: if you use any resources on the internet to develop your hash functions, make sure to cite them in your a4.c file.