CS 451
Programming Assignment #2
Imperative programming: C
Abstract Data Types

Due 2/12/16, by 6:00am sharp!

Introduction

The goal of this assignment is to get (further) acquainted with the C programming language and to gain some experience with data abstraction by constructing a hash table abstract data type (ADT) that delegates much of its functionality to a linked list.

Pervasive idea: the hash table should work for any element data type

The hash table is implemented as an array of “buckets” where each bucket is the head of linked list. If we had more time, I’d have you attempt an implementation without data abstraction. The interplay between the hash table and the linked list becomes very complex. However given a list ADT, implementing the hash table should be trivial :)

The hash table has the following parts: an array of buckets, a constructor, a destructor, three access functions, and a collection of iterators. A bucket is simply a pointer to a linked list. All values having the same “hash value” are stored in the same linked list. Hash values are computed from the data to be stored in the hash table by a hash function.

In greater detail, the constructor function $ht\text{-}initialize$ allocates space for an empty hash table, initializes its size, hash function, and comparison function, and initializes each bucket (list). The destructor function $ht\text{-}free$ frees all memory currently allocated to the hash table. The three access functions $ht\text{-}delete$, $ht\text{-}insert$, and $ht\text{-}lookup$ perform the obvious operations. Each iterator, $ht\text{-}foreach?$, (the “?” indicates the number of additional parameters the function takes) takes a hashtable, a function, and zero, one, or two additional parameters; it then calls the function with each element in the hashtable and the additional parameters.

Your implementation must use and not modify the main program and header files found on the course web page (i.e., $main.c$ and $ht.h$). You must use my main program because I will use its interface to grade your program. You must also delegate to the list ADT found on the course web page. Finally, you can modify and must use the makefile from the course web page.

What to hand in

1. Email me the source as a single attachment named $<your\ name>-a2.c$ by 6:00am on the due date. Your program’s first line must be “//This is my code.”
2. Place a copy of the executable — named $hash$ — in your linux home directory until the project is graded.

Notes

- Expect to do more thinking than coding. My solution is only fifty executable lines of code.
Security

You should secure your source code! I suggest that after logging in, you issue the following commands from your home directory

- `mkdir cs451`
- `mkdir cs451/assignment.2`
- `chmod 700 cs451 cs451/assignment.2`
- `cd cs451/assignment.2`

<download (or copy) the necessary files from the course web page>

To work on your program issue the following commands

- `cd ~/cs451/assignment.2`
- `<develop your program>`

If this is unclear please read the man pages for chmod and mkdir, or come to talk to me!!