

Homework 5: Hash Tables in C

Assigned: Friday, February 16, 2007

Due: Friday, February 23, 2007

No extensions!

Summary: In this assignment, you will implement generic hash tables.

Purposes: To give you more experience programming in C and using some of the cool C techniques.

Expected Time: Two to three hours.

Collaboration: You may work alone or you may work in a group of size two or three. I encourage you to work in groups on this assignment. You may discuss the assignment with anyone you'd like, provided you cite such discussions in your assignment.

Submitting: Email me your answer. More details below.

Warning: So that this exercise is a learning assignment for everyone, I may spend class time publicly critiquing your work.

Assignment

One of everyone's favorite data structures is the *hash table*. As you may recall, hash tables provide a mechanism by which you can create indexed tables (also known as dictionaries and maps) in which the index is a value other than a string. While there are a number of ways to implement indexed tables (dictionaries, maps), hash tables are nice because they usually provide efficient access, at the cost of a bit of extra space.

Implement and test a template-based generic hash table. That is, your hash table should let the client specify the type of the key, the type of the associated value, and the hash function used to map keys to integers. (You will be responsible for reducing the result of the hash function to the size of the table.) Your table should certainly implement the following functions.

- `VALUE get(HASHTABLE ht, KEY k)`
- `void put(HASHTABLE ht, KEY k, VALUE v)`
- `int contains(HASHTABLE ht, KEY k)`

You should provide an interactive or command-line test application for the hash table as well as a unit test.

Important Evaluation Criteria

Correct, working, solutions with a modicum of testing and a fixed-size hash table will earn a check.

Particularly elegant solutions or solutions that provide additional capabilities, such as an expanding hash table, are likely to receive a higher grade. Careful analysis of the efficacy of the hash function is also likely to earn you a higher grade. An excellent unit test may also earn you a higher grade.

Non-working or incorrect solutions will earn a lower grade.

Submitting Your Homework

Please submit this via email, using a title of CSC302 HW5.

Attach the files that you've created to that email message.