

CS69003: Computing Systems Lab I
Autumn 2008

Assignment 1

Implementation of hash tables with open addressing

Due: August 01, 2008 (Friday)

In this assignment, you are required to implement a hash table of names. Write a C program in order to solve the following parts.

Part I **(10)**

Declare a suitable C data type in order to store a hash table of s entries, where each entry is a name (that is, a character string). For the sake of simplicity, you may assume that names consist of upper case letters only and do not contain spaces. You may work with a fixed pre-determined size s (like 100) of the table.

Part II **(40)**

Read the text file "names.txt" that stores a set of names. The first line of the file indicates the count n of names to follow. In each of the lines 2 through $n + 1$, a name appears (in upper case, without spaces). Insert the names one by one in your hash table. The names in the file may be assumed to be distinct. Choose s to satisfy $n \leq s$. Resolve collisions in the hash table by linear probing. Use a special marker (like the empty string) to indicate an empty location in the hash table.

Part III **(40)**

Enter a command loop that asks the user to search/insert/delete. In case of a *search* request, your program should notify about the presence of the query name in the hash table. You should also print the number of hash table probes necessary for the search (irrespective of whether the name is found or not in the table). For an *insert* request, your program should attempt to insert a name supplied by the user. This attempt would fail if the name is already present in the hash table or if the hash table is already full. Finally, for a *delete* request, your program should attempt to delete a name supplied by the user. The attempt would fail if the name is not already present in the hash table. Use a new special marker (which cannot be a name) to indicate a deleted location in the hash table. The deleted-location marker should be different from the empty-location marker.

Part IV **(10)**

When the user enters *exit* in the command loop, your program should rewrite the text file "names.txt" by the current content of the hash table and terminate the command loop.

For your convenience, a sample file with the name "names.txt" will be supplied to you.