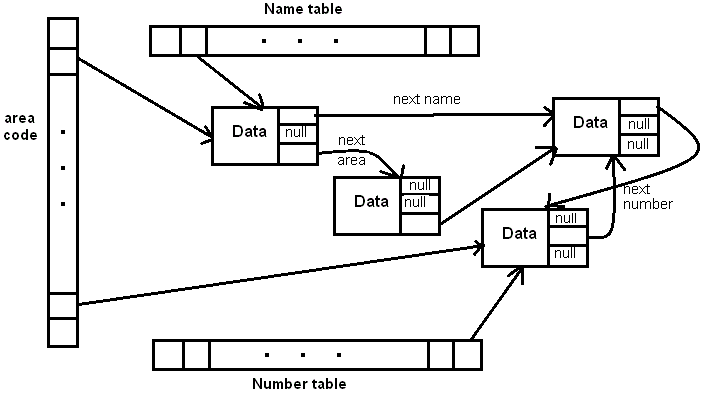
EECS 560 Project 2 Due: April 18, 2016

When a record has a single key some of its attributes cannot easily be checked. For example, if you call the phone company to get a number you must provide the name of the person. However, for billing purposes, the phone company uses the telephone number as its key. This is an example of an instance in which you may want to retrieve information based on one of two or more keys. One solution is to simply store multiple copies of the records, each organized according to a different key. However, this is inefficient and may present problems if the database is very large or if a record must be deleted since all copies would need to be removed. Instead we’ll look at a multilist structure based on two keys.

File phonebook.txt on the web page contains customer data for a telephone company. Each line is a customer’s record which includes: NAME (20 characters), ADDRESS (25 characters), NUMBER (integer, the first three digits are the area code) and CHARGES (real number with two decimal places). There is one blank space between the NAME and ADDRESS fields, and three blanks between ADDRESS and NUMBER. You are to build the phone company’s data base. You must be able to access a customer’s record using either the customer’s name or his or her phone number. You will also need to be able to give names and addresses of everyone in a particular area code. You will use hash functions to set up the database.

**Data Structures to be used:** Three hash tables: one for the names, one for the phone numbers and the third for area codes. To minimize storage space you should use a record (struct or object) that holds the data and has three pointer fields—one to point to the next name in that bucket, one to point to the next phone number in that bucket and the third to point to the next record with the same area code. The hash tables should contain pointers to records rather than the records themselves (i.e. open hashing). You may have a structure that looks something like the one below.



**Hashing functions:** For the phone numbers use division mod 53 on the last four digits of the number. For the names, take the sum of the ASCII values of the 2nd, 7th, and 13th characters of the name mod 47. For the area codes, use Knuth’s variant on division, h(k) = k(k+3) mod 113.

Note: In practice, the mod values should be changed to larger primes if the data set is so big that it leads to long chains of records with the same hash value.

Your program must be able to do the following:

1. Insert a new customer’s record by name. The name, address, phone number and charges (in that order) will be the input. Confirm that the new record was inserted.

2. Insert a new customer’s record by phone number. The phone number, name, address, and charges (in that order) will be the input. Confirm that the new record was inserted.

3. Remove a customer’s record by name. Confirm that the customer’s record was removed and print out the final bill.

4. Remove a customer’s record by phone number. Confirm that the customer’s record was removed and print out the final bill.

5. Modify a bill to reflect a payment. The input will be an amount and a phone number or an amount and a name. Output the original balance and the remaining amount due.

6. Modify a bill to reflect a charge. The input will be an amount and a phone number or an amount and a name. Output the original balance and the new amount due.

7. Print out a customer’s bill given the customer’s name.

8. Print out a customer’s bill given the phone number.

9. Given an integer, print out the names and addresses of everyone whose name hashes to that number.

10. The company bills according to the hash of the phone number. Write a function to output all phone bills for a given day. Numbers which hash to 0 or 1 get printed on the first of the month, those hashing to 2 or 3 get printed on the second, etc. (There will be some days on which no bill is printed. So, when a day of the month is input output the bills to be printed on that day including the names and addresses of the customers.

11. The phone company conducts surveys by contacting people in a particular area code. So, when an area code is input, output the names and phone numbers of all people in that area code.

Test File: To enable us to run your program, use a menu with the 11 options above plus a 12th to exit the program. Your program must be able to read from a data file with the following format:

3 //menu option

1234567 //input for that option

7

Jones Judy

6

4.56 9876543

5

10.00 Blow Joe

**. . .**

12

The name of the file that will be used is phonetest.txt and that name should be a command line argument so that it is automatically linked in when the program is run. There will also be a file for you to use in testing your program. The name of file used for testing is proj2data.txt. Feel free to add more tasks to that file.