

# LAB EXERCISE

## Hashing

### **Background:**

A larger version of the store inventory data file (*file400.txt*) will be used in this lab exercise. As you may recall, the text file consisted of two fields, *id* and *inv*. The *id* values on disk will range from 1-20000 and the *inv* fields from 1-100. There are no duplicate *id* values in the data file. Every *id* value is unique in the file and the number of store items in this problem is 400. Use this value to size your hash table.

### **Assignment:**

1. Write a program that uses a hash-coded data storage method to store the 400 items in (*file400.txt*). You can start with an earlier program such as the binary search lab to speed up your work.
2. The method of dealing with collisions should be the dynamic linked list version.
3. You should develop your own hashing scheme to take the key field (*id*) and determine the correct address.
4. Test your new data structure and algorithm with some sample searches. The program should prompt you for an *id* value and return the *inv* amount or a message that the *id* does not exist. Your instructor will specify some sample *id* values to test out.
5. Your program must analyze the efficiency of your hashing scheme by determining the following statistics about your hash table:
  - a. The % of **null** pointers in the hash table.
  - b. The average length of linked lists.
  - c. The longest linked list in the hash table.

After seeing these results, you might want to try to improve upon your hashing scheme if the number of collisions is excessive.

### **Instructions:**

1. Display your source code and a run output of the following results:
  - a. The searches
  - b. The statistical analysis of the hash table.