Computer Information Science 430
M. Dixon
Assignment 1 - Hashing
Project Specification

In many applications, it is desirable to access data in a sequential manner. Examples of this can be found in payroll and billing processing. However, certain applications, like airline reservations, require **immediate, random** access to data.

This assignment introduces the concept of ***hashing***. Hashing refers to the process of generating a unique address or index for a data value or object. Typically, this unique address or index is used to determine the object's relative position within a data structure or disc file.

Processing Requirements

"DATAIN" is a text file which contains 59 records whose format is KEYDATA, where KEY is a character (string) field of 10 characters and DATA is a character (string) field of 20 characters (see example below). This file will be provided to you. Read "DATAIN" and hash each record using a hash function, later specified, into the memory data structure. Collision resolution handling is specified later.

K E Y (10 bytes)	D A T A (20 bytes)

```
VEN-TAL INWALSH SANTA CLARA CA
TAXAN CORPCITY OF INDUSTRY, CA
ORCHID CORWESTINGHO FREMONT CA
SSISOFTWARCENTER ST. OREM UTAH
```

The memory data structure, called a ***hash table***, consists of a primary and overflow area. The primary are has 20 buckets, each bucket containing 3 hash slots, an overflow pointer and count fields. The overflow area contain an additional number of buckets dedicated to handle collisions. Bucket zero's overflow pointer field may be used to point to the next available overflow bucket in the event of a collision.

Bucket n, Slot 1	K E Y	D A T A	
Slot 2	K E Y	D A T A	
Slot 3	K E Y	D A T A	
	Overflow Pointer		Count

Hashing (contd)

A ***collision*** occurs when two or more data keys hash to the same bucket. If a collision occurs and there is a vacant slot in the primary area bucket, the record is stored in the first available slot. If the primary area bucket is full and an overflow bucket has been allocated, store the record in the next available slot within overflow area bucket. If no overflow bucket has been allocated, allocate one (link the primary area bucket to its overflow bucket using the pointer field).

After hashing all records contained within the file "DATAIN", write the memory data structure to a file called "HASHTABLE" and display the contents of all hash buckets in the report format shown below.

Hash Table
Verification Report
Before | After Restoration

Bucket 1
    Slot 1:
    Slot 2:
    Overflow Pointer: xx

..
..

Bucket n
    Slot 1:
    Slot 2:
    Overflow Pointer: xx

The second phase of the assignment involves restoring the hash table from a disk file and searching for data keys found in file 'SEARCH' which contains 20 records. This file will be provided to you.

Sample Entries ("SEARCH" file)

MICROWAY C
TATUNG CO.
DYSAN CORP
DIGITAL RE

First, restore the memory data structure using the file "HASHTABLE" and produce the restoration report (format shown above). Next, read records from the file "SEARCH" and hash the key. Simulate the retrieval of a matching record from the data structure by producing the report shown on the next page.

Search and Retrieval
Transactions

Search Key	Bucket/Slot	Record (show data image here)
DYSAN CORP	12/2	PAT H SANTA CLARA CA
DIGITAL RE	4/1	GARDEN C MONTEREY CA
FUNK SOFTW		Record not found

The final phase of the project consists of computing total and average chain (collision) lengths of the hash function. Compute the total length for each primary bucket and the average of all primary buckets. For computing the average, exclude buckets having zero collisions. Use a report format similar to the before/after restoration report to report total length for each primary bucket.

Hash Functions

Hash functions are a ***key to address transformation*** and their efficiency is usually dependent upon the data set being "hashed". However, the simplest hash algorithm is also among the most efficient. For this assignment, use the following:

Intermediate Value = Ord(key[2])+Ord(key[4]+Ord(key[6])

Hash index = (Intermediate Value) mod table size
where table size is number of primary area buckets.

Thus, this assignment consists of following major tasks:

1) Hashing the records found in 'DATAIN'(ftp://165.196.201.8/dixon/cisp430) to a memory data structure.

2) Writing the memory data structure to disc and later, restoring the structure.

3) Displaying the contents of hash buckets before and after restoration.

4) Search of the memory data structure for key values found in file 'SEARCH'(see ftp above).

5) Computing hash efficiency statistics.

Submit the following items for grading:

1) A copy of the source file.

2) Copies of all reports produced by the program; there are 3 reports: the hash table contents before and after restoration from disc, and the search/retrieval report.

3) Any project documentation you wish to include.