# *Homework 5*

*100 Points*

# *Hashed Tables*

Write a menu-driven program that implements a simple hashed table based database.
The program reads data from a text file and inserts them into the hashed table. Collisions will be resolved using linked lists. Once the hashed table has been built, present the user with a menu:

**S** – Search by a unique key
**D** – Display list: display the contents of the hash table as shown on the next page.
**P** – Print the hashed table: show the index and indent synonyms, as shown on the next page.

**T** – Show statistics, such as number of collisions, load factor, number of linked lists, longest linked list, average number of nodes stored in linked lists, etc.
**M** - Show Menu, and
**Q** – Quit.

Input File: Same as in Homework 4 (old movies). The key is to be the movie title.
The hashed table is to be dynamically allocated. Run the program twice first with size 10, then size 31.

Define and use a hash function of your choice. Provide at least one test case for each option, except for search, for which provide more test cases. Send the output to the screen. When done, copy and paste it to `Hash_Output.txt`.

## Grading

| | |
|---|---|
| 1. Build hashed table | – 20 |
| 2. Search | – 20 |
| 3. Display | – 15 |
| 4. Print | – 20 |
| 5. Statistics | – 20 |
| 6. main() | – 5 |

OUTPUT FORMAT

**D** – Display list: display the contents of the hash table as shown below:

```
ItemM
ItemB
ItemQ
ItemA
ItemD
```

**P** – Print the hashed table: show the bucket number, the empty slots and indent synonyms, as shown below:

```
Index 0: ItemM
                ItemB
                ItemQ
Index 1: // empty
Index 2: ItemA
                ItemD
```