**ECE 370 Fall 2017**

## Programming Assignment 3

Assigned 10/9/2017
Due 10/27/2017

### Build two hash tables

Assume that in the file, **a3.txt**, there are no more than 50 names (entries). Each entry is less than 15 chars. If a state name is longer than 14, your program needs to truncate it. The data format is as follows:

NewYork
Connecticut
Michigan
Ohio

1. **hash table 1**

**Hash and Linear Probing:** You are to construct a closed hash table of total capacity of hosting **100** entries. Using ***linear rehash (or linear probe: one step increment)*** with the following hash function to insert each string ***inString*** from the input in **a3.txt** into the hash table:

**h(inString) = (inString[0] + inString[last] + i) mod 100; i = 0, 1, 2, ….**

inString[k] is the Ascii value of char inString[k].

Record the cumulative number of steps (i.e., the number of hash table slots examined for available slot) for every string entered; **this number must be printed immediately following each state name in the table**. Print the resulted hash table on the screen: **4** state names each line; if the bucket entry is NULL, print **15 underscore "_"** that are assigned for each entry. Enumerate each line with the first entry of the four; e.g. for first line, the line number is 01. Each entry should be printed as 15 chars. For a name less than 15 chars, use spaces to make up. For example, if NewYork has **four** collisions, and NewJersey has **twelve** collisions:

```
NewYork        04  NewJersey      12
```

2. **hash table 2**

**Repeat part 1, with Quadratic Probing**

**h(inString) = (inString[0] + inString[last] + $i^2$) mod 100; i = 0, 1, 2, ….**

.

3. **Comparison:** After the hash tables built, do a search for each of these two tables. The search is on each and every entry in a3.txt, and count on the number of comparisons for searching each entry. Print out the total number of comparisons in the **hash table 1** and **hash table 2**.