**CS 248 – Object-Oriented Programming and Data Structures**

**HW8: 100 points**

**Objective:**

The objective of this homework is to:

1. Understand the concept of Hashing
2. Apply programming skills to implement a solution to a specific type of collision resolution scheme (double hashing)

**Problem Statement:** You need to implement a solution code for the double hashing collision resolution scheme. The description of each function that needs to be implemented is given in the starter code

**Use the following hash functions:**

**Hash function1:**

**h1(k)= key mod TABLE\_SIZE**

**Hash function2:**

**h2(k)= 1 + (key % PRIME)**

**nextindex = (h1(k) + i\*h2(k)) mod TABLE\_SIZE**

**Input Format:** You need to read from a file named **input.txt (make sure you use exactly this name)**. First line of the file contains 3 integers (space separated): numbers of keys to be inserted (N), number of keys to be deleted (D), and number of keys to be searched for (S). Second line contains N integers (space separated), which are keys to be inserted in the hash table. Third line contains D integers (space separated), which are keys to be deleted. Fourth line contains S integers (space separated), which are keys to be searched in the hash table.

**Note: Call the display function after inserting N keys. Then call again after deleting D keys.**

**HW Grading:**

1. 10% - Follows the instructions given in the HW (correct filename, correctly calling display function after insertion and deletion etc.)

1. 10% - No warnings
2. 30% - Compiles without errors
3. 50% - Correct implementation of all functions

**Sample input/output:**

**For TABLE\_Size = 13, PRIME=11 (use these values in your submitted code)**

**Sample Input from file:**

6 3 3

79 69 72 50 98 14

79 98 50

14 69 79

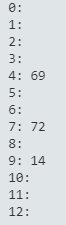
**Sample Output:**

**After insertion:**

Current elements in hash table:



**After deletion:**



**After Search:**

14 found at index 9

69 found at index 4

79 not found

**Bonus (10 pts):** Implement the Rehash function. Define a constant named LOAD\_FACTOR. If the load factor is greater than 0.8, double the size of current TABLE\_SIZE and redistribute the current hsahtable keys in to the new hashtable.