BLG335E, Analysis of Algorithms I, Fall 2013
Project 3
Handed out: 5.12.2013
Due: 22.12.2012, until 11.30 PM
Linear Probing, Double Hashing, Universal Hashing

1. **(15 points)** Implement a m-sized hash table that uses linear probing strategy in order to store integer key elements by using the following hash function. Each slot of your hash table can only contain an integer value.

$$h(k, i) = (k + i) \bmod m \text{ and } i \in [0, m - 1]$$

2. **(15 points)** Implement a m-sized hash table that uses double hashing strategy in order to store integer key elements by using the following hash function. Each slot of your hash table can only contain an integer value.

$$h(k, i) = (h_1(k) + i * h_2(k)) \bmod m \text{ and } i \in [0, m - 1]$$

$$h_1(k) = k \bmod m \text{ and } h_2(k) = 1 + k \bmod (m - 1)$$

3. **(20 points)** Implement a m-sized hash table that uses universal hashing strategy in order to store integer key elements. Details of the universal hashing strategy is given below.
   Decompose key k into 3 digits (in this case $r = 2$) each with a value in the set $[0, m - 1]$.
   Therefore, $k = [k_0, k_1, ..., k_r]$ where $0 \leq k_i < m$
   By using the randomized strategy, pick $a = [a_0, a_1, ..., a_r]$ where each a_i is chosen randomly from $[0, m - 1]$.
   Then define the hash function as follows;

   $$h_a(k) = \sum_{i=0}^{r} a_i k_i \bmod m$$

   Example decompositions are given as follows:
   $k = 154896356 \rightarrow k_0 = 154, k_1 = 896, k_3 = 356$
   $k = 4896356 \rightarrow k_0 = 4, k_1 = 896, k_3 = 356$
   $k = 96356 \rightarrow k_0 = 0, k_1 = 96, k_3 = 356$

4. **(10 points)** You are given another input file (search.txt) that also contains key values. Write a function that searches given set of keys in the hash table that you implemented in (1). Your function should take the file name that contains the key values (search.txt).

5. **(10 points)** Write a function that searches given set of keys in the hash table that you implemented in (2). Your function should take the file name that contains the key values (search.txt).

6. **(10 points)** Write a function that searches given set of keys in the hash table that you implemented in (3). Your function should take the file name that contains the key values (search.txt). Remember that you should search by using the same hash function that you used during inserting procedure in (3).

7. **(20 points)** Number of collisions when inserting/searching an item k is defined as follows:

   initiliaze *collisions = 0*

   Compute $h(k,i)$, if $h(k,i)$ is not empty, then increment *collisions* by 1

   Using the definition above, insert all the elements in "insert.txt" to the hash tables, compute the number of collisions for each strategy and fill in the table with the number of collisions occured for each strategy and m value. Similarly, search all the elements in "search.txt" and compute the number of collisions for each strategy. Comment on the results.

	Insertion		
	Linear	Double	Universal
m = 1327			
m = 2657			

	Search		
	Linear	Double	Universal
m = 1327			
m = 2657			

Detailed Instructions:

- All your code must be written in C++ using ***object oriented approach*** and able to compile and run on linux using ***g++***.

- Do not use additional libraries such as STL.

- After the execution of your program, collusion values for each strategy and m value should be printed out to the screen.

- Submissions will be done through the Ninova server. You must submit your source code file (a single file with .cpp extension) and a softcopy report (.pdf).

If you have any questions, please contact Res. Asst. Doğan Altan via e-mail (daltan@itu.edu.tr).