CSCI 335
Third assignment
Total: 100 points

Due Tuesday Oct. 30
Please, follow the blackboard instructions on writing and submitting programming assignments
We will not debug your assignment. It should run correctly to receive credit.

Hash-map (100 points)

Implement the most efficient version of the "word puzzle" algorithm of section (the one shown in Figure 4.73) using your own hash map. Create your own *templated* hash-map class that only supports lookup (find), insert and delete (you can use lazy deletions) operations. You should also provide an operation that visits all the elements in the hash-map (in the way they are stored in the hash table). The hash-map will now store a key with an associated value. You don't have to implement iterators for your hash-map implementation, but you can if you wish.

Your executable should run as follows:

```
./test_hash_map <word_file.txt> <hash_type>
```

where `<hash_type>` can be quadratic or double hashing.

For example

./test_hash_map words.txt chaining

will use a separate chaining hash-map implementation.

The *test_hash_map* executable should do the following

(a) Create the final map (the one returned by the algorithm in figure 4.73)
(b) Report how much time did this map creation took (code to use for counting time a piece of code takes to execute is included).
(c) Prompt for a word to be given by the user.
(d) Printout the list of associated words.
(e) Prompt for a second word to be given by the user.
(f) Printout the list of associated words.

A starting piece of code is included, with the implementation of quadratic probing. Also note that the code provided for hashing saves keys, into the table. You need to modify that code, in order to store key-value pairs. Start with a full implementation of quadratic

probing. Then implement double hashing. That would require a small change in your quadratic implementation.