Top-20 Training Program
(Set & Map)

Apply the problem solving techniques discussed in class to solve the following problems.

Problem1: Spell Checker

Given an input file and dictionary of words, write an efficient program to display the words that are spelled incorrectly.

Problem2: Bidirectional Symbol Table

Create a data structure that supports following operations in fastest time:

put(key, value)
getByKey(key)
getByValue(value)

This kind of datastructure is very useful in DNS lookup and DNS reverse lookup i.e., domain name to IP resolution and viceversa.

Problem3: Inverted index of web

Given a list of web pages, create an inverted index for the words of those web pages i.e., associate with each word a list of web pages in which that word appears. Write a program that reads in a list of web pages, creates an inverted index, and support multi-word queries. That means, it must output the list of web pages that contain at least one occurrence of each of the query words.

Problem4: T9 Dictionary

Implement T9 mode of mobile input. Assume following mappings for digits: 1 – ABC 2 – DEF 3 – GHI 4 – JKL 5 – MNO 6 – PQRS 7 – TUV 8 – WXYZ

Function prototype:

```
void findT9suggestWords(String input, String dictionary)
//input contains the digits pressed in order
```

Problem5: LFU cache design

Create a cache with fixed size that supports the following operations:

boolean put(key, value) - add the item onto cache if it's not already present. If cache is full then remove the least frequently used element and then add this element.

value get(key) – return the value associated with key if present otherwise return null

Top-20 Training Program
(Set & Map)

void clear() – Clear the contents of cache

void display() – Display the contents of cache

Hint: In class we designed LRU cache with DLL and HashMap combination. Think of what way do u organize DLL elements for frequency based access and ofcourse, HashMap will be used for indexing.