Shiv Chandra Kumar **Program 3**

Design decisions and strategies for Implementation:

1. I implemented Hash table first as suggested in the plan of attack. I took N = 69997(prime number from the list given) as my table size. I took R = 69931(a saller prime number than table size) for double hashing.
2. i used first hashing function as h1(x) = key(mod)N. Here key value is calculated from the pseudocode provided in the problem set. Double hashing function h2(x) was taken as h2(x) = R - key(mod)R
3. formula used for double hashing is h1(x)+i(h2(x) where i is number of attempts for double hashing in order to avoid collision.
4. I used the concept of singleton function for Hash class in order to make sure that only one instance of the class can be created during the entire operation by wrapper class Dictionary.
5. I implemented 3 methods for Adding new words, removal of words and looking for availability of the word. among these first two are exclusively used only by Dictionary in case of adding new words or removal of archaic words.
6. I implemented a Dictionary class as a wrapper class for hash class. All the methods inside Hash class can only be called via an object of Dictionary class.
7. Dictionary class has same methods as Hash class and are actually calling methods of Hash class. methods here are add, remove and lookup(same as Hash).
8. In the end i implemented A spellchecker Class which has methods for finding correct alternatives for mis spelled words.
9. As mentioned I have used 3 techniques out of the 5 given in program sheet.
10. Techniques i have used are
    1. A. CheckSwap
    2. CheckDelete
    3. CheckSplit
11. As i have implemented only three out of the 5 algorithms, as suggested, we will not find suggestion for some incorrect words in final output. Those words requires CheckReplace and CheckInsert technique which has not been implemented.
12. I am using all three private methods to implement above 3 algorithms. I am using a single public method to call all these private methods of algorithm as well as for printing final output.
13. Finally i am reading file character by character until i get a complete word and check using lookup method whether it is there in dictionary or not. If it is not there i am calling correcting measure using above mentioned public method in spellchecker class.
14. all three algorithms are being checked one by one and upon finding a word for replacement i am pushing that word into an array which i have maintained as a private variable of Spellchecker class.
15. At the end i am sorting the words in array in ascending order to print the suggestion in that order.
16. finally after printing the suggestion for a word i am clearing memory of array and moving on to next word to check its existence.