Assignment # 3
Data Structures & Algorithms

Due Date: 27-10-2021 (11pm) **Marks: 20**

Submission Guidelines:

- Submit your assignment only on google classroom within deadline.
- Late submissions of upto 15 mins are allowed with a penalty of 20% on google classroom.
- Add a comment section atop of your code files, mentioning your name roll number etc, and a bit about what you did in this code. Follow the detailed guidelines shared by the TA.
- Name/rename your submission files as "Section-roll#-Assig#-filenmae.cpp", e.g., "A-20i0899-A2-main.cpp".
- Zero marks will awarded in following cases
    - Code with build errors
    - Submitting assignment through email, slate, or any other way other than the google classroom
    - Submitting after the late deadline

Spell Checker using Hashing

You are provided a repository of English words, which contains almost all the words used in the text. You need to insert all these words from the repository into the hash table using chaining method.

After you have created the hash table, your program is ready to do the spell checking. For this purpose, a user will input some text into an input.txt file to check for any spelling mistakes. Your program should search for each word of the input text from the hash table to see if it's there or not. If the word is not found, then the missing word should be capitalized to make it clearly visible. After all the words have been searched, the program should rewrite the input text in the same file with those words capitalized which were not found in the hash table.

If the input text contains words with characters other than the English alphabets, then you should not try to search such words. To search a word from the hash table, it first needs to be converted to lowercase before the searching, because all the words in the hash table are lowercase.

Note that the repository contains ten thousand words in lowercase, therefore select the array size accordingly. It is recommended to select a prime number between 9,000 and 10,000

Performance:
In this assignment, insertion is carried out only at the start, and then later on the program will only be searching words. As we know that searching is O(1) in hashing, given a good hash function. So hashing is the natural choice for our problem.
A good hash function will minimize collisions, and will hash the data to an empty slot. You need to work on the hash function to minimize the collisions. **To check out, how good your hash function is, you will need to find out the total number of occupied array slots. Bigger the number, the better it is. Ideally this number can equal the number of total words.**

Program Output:

- Text of input fie with wrong spelled words converted to uppercase to replace the text in the input file
- Print number of Non-empty Array slots of hash table

HashChain Class:
To implement hashing, you need to implement the following class design.

HashChain
Table: *list<> size: int
HashChain(int): ~HashChain(); insert(item): void search(key): bool