BASIC SPELL CHECKER USING A HASH TABLE IN C LANGUAGE

Introduction:

The above code is a C program that implements a basic spell checker using a hash table. The program loads a dictionary of words from a text file and stores them in a hash table. The program then takes input from the user, checks the spelling of the input word by looking up the hash table, and returns whether the word is spelled correctly or not.

Algorithm:

The spell checker algorithm used in this program can be summarized in the following steps:

1. Load a dictionary of words from a text file into the hash table.
2. Take an input word from the user.
3. Calculate the hash of the input word.
4. Check if the input word is present in the corresponding index of the hash table.
5. If the word is present in the hash table, it is spelled correctly. If not, it is spelled incorrectly.
6. Repeat steps 2-5 until the user decides to exit the program.

Working Procedure:

- The program starts by including the necessary headers and defining the constants such as MAX_WORDS and MAX_WORD_LEN.
- It then defines a struct called node which will be used to create linked list for hash table and a hash table of nodes.
- The program then defines a function called hash which takes a word as an input and returns an index for the hash table by using a simple hash function that multiplies the ASCII value of each character by a prime number and then takes the modulo with the size of the hash table.
- The addWord() function takes a word as an input, calculates its hash, creates a new node, and adds the word to the corresponding index in the hash table.
- The checkSpelling() function takes a word as an input, calculates its hash, and checks if the word is present in the corresponding index of the hash table.

- The program then opens the dictionary file and reads it, word by word and adds it to the hash table using the addWord() function.
- The main function takes input from the user and checks the spelling of the word using the checkSpelling() function.

Sample Output :

Here is an example of the output generated by the program when the user inputs the word "cat"

```
Enter a word: cat
The word is spelled correctly
```

And when the user inputs the word "catt"

```
Enter a word: catt
The word is spelled incorrectly
```

It's important to note that the output will vary based on the words present in the dictionary file, also the hash function used in this example is a basic one and it may have collisions when used with large datasets. In addition, the program only checks for exact matches of the input word in the dictionary and doesn't take into account variations such as plurals or verb conjugations. This is just a basic implementation, a more advanced spell checker would include additional functionality such as handling variations and suggestions for misspelled words.

Conclusion:

The given code is a basic implementation of a spell checker using a hash table in C. It loads a dictionary of words from a text file and stores them in a hash table, and then takes input from the user, checks the spelling of the input word by looking up the hash table, and returns whether the word is spelled correctly or not. The algorithm used in this program is simple and efficient, but it has some limitations such as hash collisions and it only checks for exact matches. More advanced spell checkers would include additional functionality such as handling variations and suggestions for misspelled words. It's a good starting point for understanding the basic concepts of spell checkers using hash tables.