Daffodil International University
Faculty of Science & Information Technology
Computer Science & Engineering (CSE)

Project Details:

Project Name	Tiny Spell Checker
Language	C++
Team Name	Red Chilies
Submitted to	Zakia Zaman
Submission Date:	23.08.2016

Team Members Info:

Name	ID
Iftieaq Murshed	153-15-6572
Kazi Ariful Islam	153-15-6567
Rabeya Kamal Surovi	153-15-6578
Mukul Serker	153-15-6658
Juwel Ahmed	153-15-6677

Abstract

Tiny Spell Checker is a simple spell checker application which takes input from user and check each and every word from it. If there are any misspelled word, the speller checker will generate an alert and collect some autosuggestion and ask user to input a corrected version of the word. And finally it will output the corrected version of the input stream.

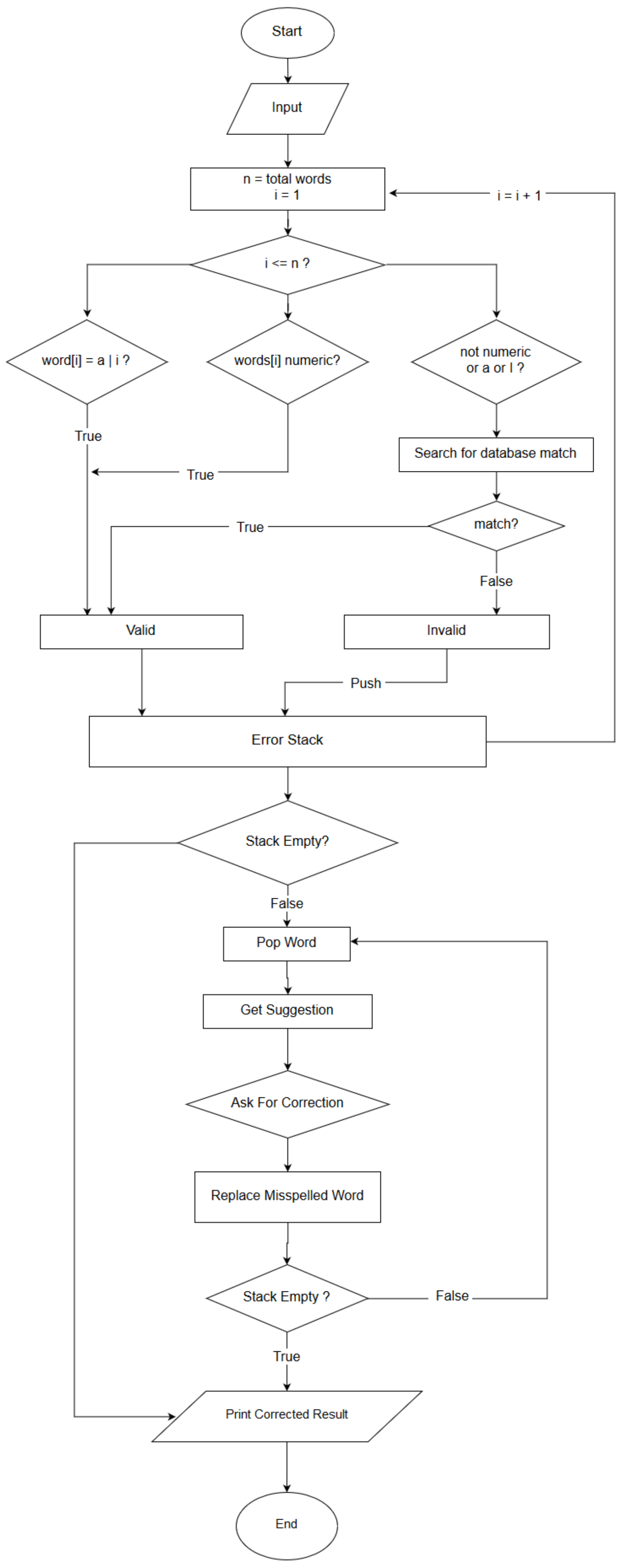
Table of Contents

Introduction

Occasionally you need to check spelling for your blog, email or anywhere and you don’t have any word processor software for that. This is when Tiny Spell Checker becomes handy. It is a light weight C++ application that allows you to easily and quickly check and correct your misspelled words and also allow you to copy the corrected output so that you don’t have to find each wrong word from you content and replace all of them manually.

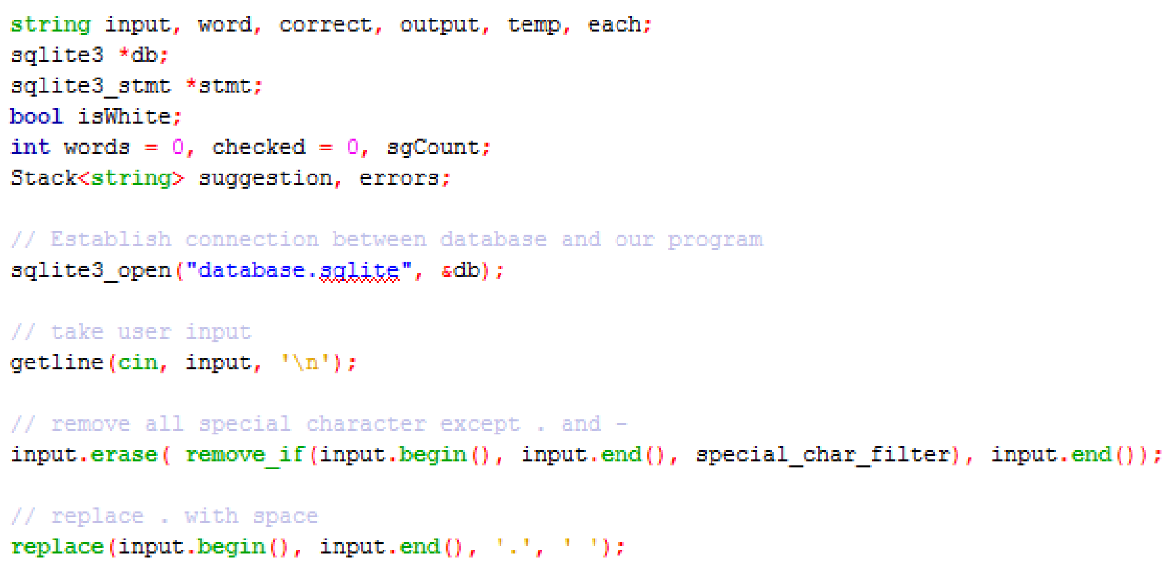
Tiny Spell Checker also comes with an auto suggestion feature which will suggest you the correct spelling of a misspelled word.

Flow Chart



Start
Input
n = total words
i = 1
i = i + 1
i <= n ?
word[i] = a | i ?
words[i] numeric?
not numeric or a or I ?
True
Search for database match
True
True
match?
False
Valid
Invalid
Push
Error Stack
Stack Empty?
False
Pop Word
Get Suggestion
Ask For Correction
Replace Misspelled Word
Stack Empty ?
False
True
Print Corrected Result
End

Establish Database Connection



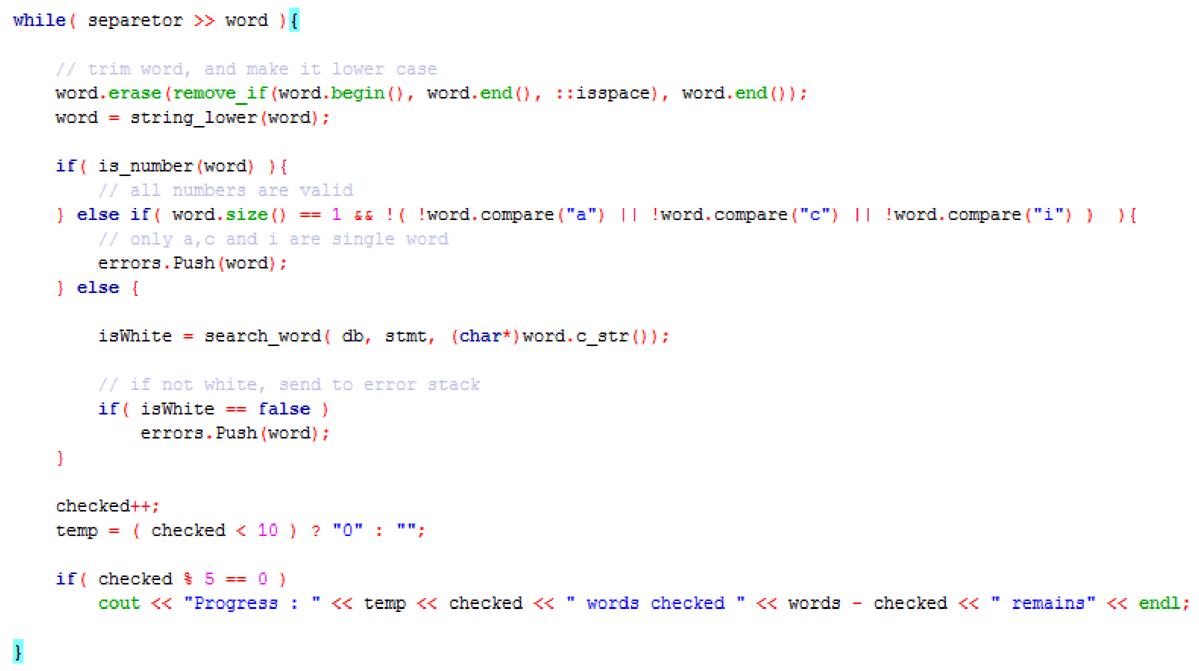
```
string input, word, correct, output, temp, each;
sqlite3 *db;
sqlite3_stmt *stmt;
bool isWhite;
int words = 0, checked = 0, sgCount;
Stack<string> suggestion, errors;

// Establish connection between database and our program
sqlite3_open("database.sqlite", &db);

// take user input
getline(cin, input, '\n');

// remove all special character except . and -
input.erase( remove_if(input.begin(), input.end(), special_char_filter), input.end());

// replace . with space
replace(input.begin(), input.end(), '.', ' ');
```

In this project we are using SQLite Database management system. Establishing connection with SQLite database is very simple. In the snippet we use the sqlite_open function to connect our program with our database. We already have a SQLite database which contains more than three hundreds of thousands English words.

Later we use getline function to take input from user and it will take input until user press the Enter button. And the we do some replacement in the given string because there are lots of special character in the world. And it's quite impossible to verify all word with any possible special character. So it's a wise step to erase them at the first place.

Another important point is replacing the full stop with space because we are separating each word from the given string based on the position of space between each word. And when someone write a full stop, it means he/she is going to start a new line and it may not contain a space at the very beginning.

Primary and Secondary Verification



```
while( separetor >> word ){

    // trim word, and make it lower case
    word.erase(remove_if(word.begin(), word.end(), ::isspace), word.end());
    word = string_lower(word);

    if( is_number(word) ){
        // all numbers are valid
    } else if( word.size() == 1 && !( !word.compare("a") || !word.compare("c") || !word.compare("i") )  ){
        // only a,c and i are single word
        errors.Push(word);
    } else {

        isWhite = search_word( db, stmt, (char*)word.c_str());

        // if not white, send to error stack
        if( isWhite == false )
            errors.Push(word);
    }

    checked++;
    temp = ( checked < 10 ) ? "0" : "";

    if( checked % 5 == 0 )
        cout << "Progress : " << temp << checked << " words checked " << words - checked << " remains" << endl;

}
```

Our main algorithm is separate each word and compare each of them with our database result. But before that we may do some primary test in order to reduce time complexity. For example, if user type an integer then we don't need to check the spelling for that because all integers are valid word and also if a word is a single character and it's not "A", "C", or "I" then it must be a misspelled word because there are only three single character words in the dictionary. So if we run some basic test before database query, we may save some valuable time ☺.

If the word doesn't pass the primary test condition, the we have no choice but send it for database verification. Here the search_word function will do that for us. It will return true if the word exists in our database, otherwise false.

So if the word doesn't exist in our database, we will push the word into our error stack. This process will continue until pointer reached to the end of the string.

Mr. Suggester

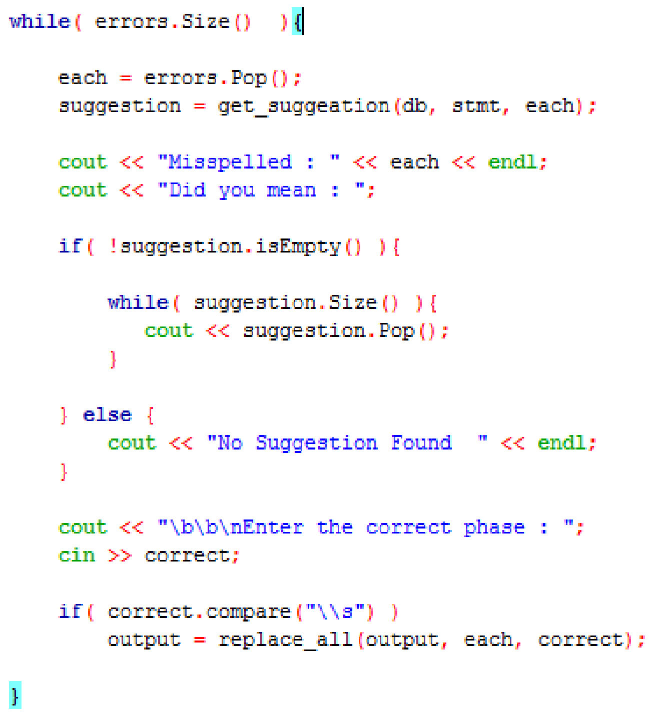
```cpp
Stack<string> get_suggeation( sqlite3 *db, sqlite3_stmt *stmt, string str ){

    string strTemp;
    Stack<string> output;

    strTemp = str;

    for( int i = str.size() - 1; i >= 0; i-- ){

        strTemp[i] = '%';
        sqlite3_prepare_v2(db, "select * from words where word LIKE ? COLLATE NOCASE; LIMIT 5", -1, &stmt, NULL);
        sqlite3_bind_text(stmt, 1, strTemp.c_str(), -1, SQLITE_STATIC);

        // if search found something, loop through it
        while ( (rc = sqlite3_step(stmt)) == SQLITE_ROW) {

            string res = string(reinterpret_cast<const char*>( sqlite3_column_text(stmt, 1) ));
            output.Push( res );

            if( output.Size() > 15 )
                break;
        }

        sqlite3_finalize(stmt);
        strTemp = str;
    }

    return output;
}
```

A common need in spell checker applications is suggesting query terms or phrases based on incomplete or misspelled user input. This is exactly what Mr. Suggester do. When you type a wrong or incomplete phase, this function will perform various search query to generate an approximate correct result.

One of the most delicious part a database management system is it's search pattern. We know C++ has very poor regular expression support so we can't be happy with that. But with SQLite and SQL we can perform some really quick and effective search using "%" sign. If you put a % sing in any position of a string and execute a search command, the SQL will detect it as "Anything" on that place and perform query.

For example, searching with "B%%K" will bring both "BOOK" or "BANK" or "BOND" and so on.

Error Handler



```
while( errors.Size()   ){

    each = errors.Pop();
    suggestion = get_suggeation(db, stmt, each);

    cout << "Misspelled : " << each << endl;
    cout << "Did you mean : ";

    if( !suggestion.isEmpty() ){

        while( suggestion.Size() ){
           cout << suggestion.Pop();
        }

    } else {
        cout << "No Suggestion Found  " << endl;
    }

    cout << "\b\b\nEnter the correct phase : ";
    cin >> correct;

    if( correct.compare("\\s") )
        output = replace_all(output, each, correct);

}
```

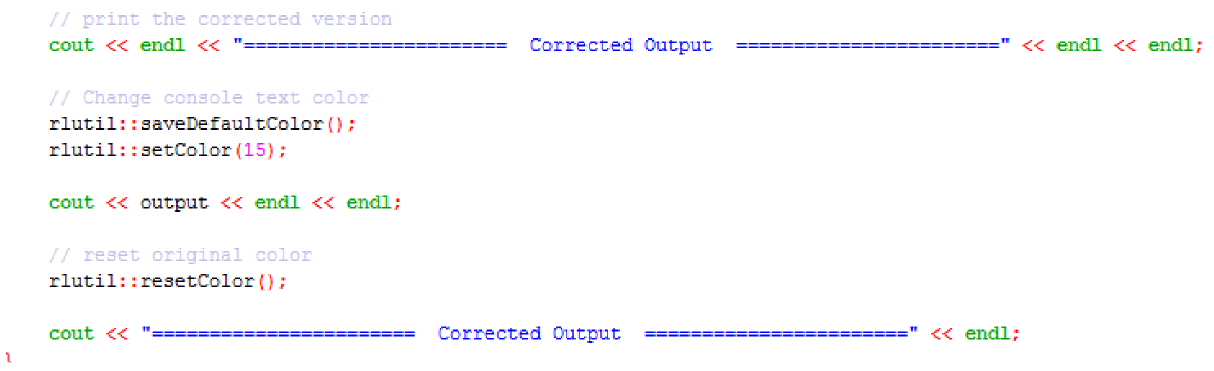
So we discussed about the word verification and auto suggestion generators. Now it’s time to pop the error and replace them with their corrected version.

At the beginning we push all misspelled words into an error stack. Now we will perform a loop which will continue until the stack becomes empty. And on each time we will send a request to Mr. Suggester to generate some suggestions for us.

If we have any suggestion, we will print that out into the console otherwise we will print “No Suggestions Found”. And then we will ask user to pick one of the suggestion or type something from this own.

And finally we will replace all occurrence of the word from our input string. We also put a skip command “\s” which will skip the replacement for that word.

Happy Ending



```
    // print the corrected version
    cout << endl << "======================  Corrected Output  ======================" << endl << endl;

    // Change console text color
    rlutil::saveDefaultColor();
    rlutil::setColor(15);

    cout << output << endl << endl;

    // reset original color
    rlutil::resetColor();

    cout << "======================  Corrected Output  ======================" << endl;
```

We are almost done here, we collected all wrong words, we generate suggestions for them, we ask user for their corrected version and also we replace the misspelled words with the correct one.

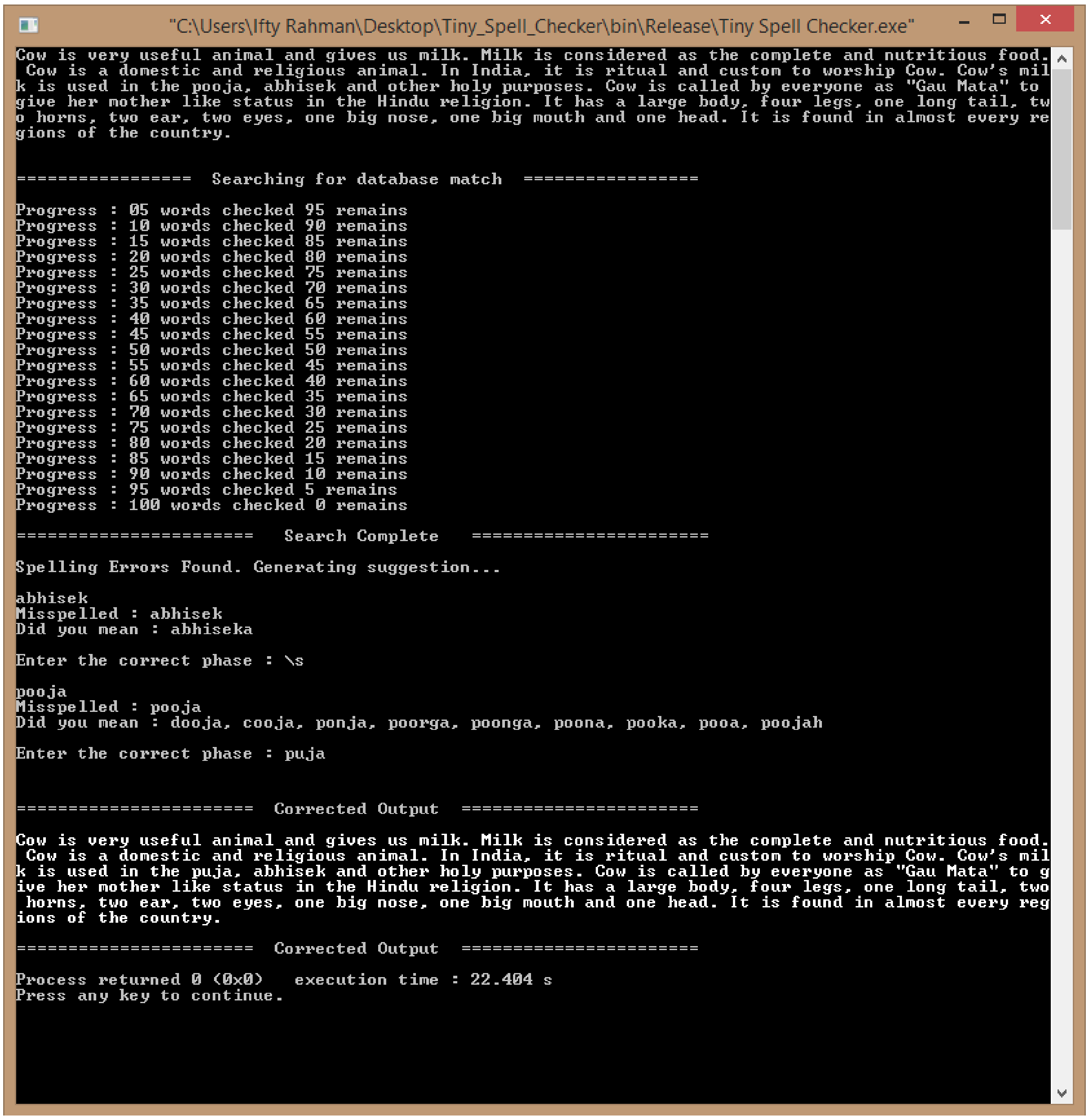
Now we have only one thing remains, printing the corrected output. But before that we will do some fun with console text color to make our output colorful and also beautiful.

For doing that, we are using a third partly C++ library called "rlutil". It is free and open source. The source code can be found in GitHub.

The uses of rlutil is simple. First we save the default color into a property so that we can restore it later. Then we use setColor method to change color. There are total 16 colors available, starting from 0. Just pass the color index as a parameter and rlutil will do the rest.

After printing output, we use resetColor function to reset the console because we only want to highlight our output, nothing else.

Sample IO



"C:\Users\Ifty Rahman\Desktop\Tiny_Spell_Checker\bin\Release\Tiny Spell Checker.exe"

```
Cow is very useful animal and gives us milk. Milk is considered as the complete and nutritious food.
 Cow is a domestic and religious animal. In India, it is ritual and custom to worship Cow. Cow's mil
k is used in the pooja, abhisek and other holy purposes. Cow is called by everyone as "Gau Mata" to
give her mother like status in the Hindu religion. It has a large body, four legs, one long tail, tw
o horns, two ear, two eyes, one big nose, one big mouth and one head. It is found in almost every re
gions of the country.


=================  Searching for database match  =================

Progress : 05 words checked 95 remains
Progress : 10 words checked 90 remains
Progress : 15 words checked 85 remains
Progress : 20 words checked 80 remains
Progress : 25 words checked 75 remains
Progress : 30 words checked 70 remains
Progress : 35 words checked 65 remains
Progress : 40 words checked 60 remains
Progress : 45 words checked 55 remains
Progress : 50 words checked 50 remains
Progress : 55 words checked 45 remains
Progress : 60 words checked 40 remains
Progress : 65 words checked 35 remains
Progress : 70 words checked 30 remains
Progress : 75 words checked 25 remains
Progress : 80 words checked 20 remains
Progress : 85 words checked 15 remains
Progress : 90 words checked 10 remains
Progress : 95 words checked 5 remains
Progress : 100 words checked 0 remains

=======================   Search Complete   =======================

Spelling Errors Found. Generating suggestion...

abhisek
Misspelled : abhisek
Did you mean : abhiseka

Enter the correct phase : \s

pooja
Misspelled : pooja
Did you mean : dooja, cooja, ponja, poorga, poonga, poona, pooka, pooa, poojah

Enter the correct phase : puja


=======================  Corrected Output  =======================

Cow is very useful animal and gives us milk. Milk is considered as the complete and nutritious food.
 Cow is a domestic and religious animal. In India, it is ritual and custom to worship Cow. Cow's mil
k is used in the puja, abhisek and other holy purposes. Cow is called by everyone as "Gau Mata" to g
ive her mother like status in the Hindu religion. It has a large body, four legs, one long tail, two
 horns, two ear, two eyes, one big nose, one big mouth and one head. It is found in almost every reg
ions of the country.

=======================  Corrected Output  =======================

Process returned 0 (0x0)   execution time : 22.404 s
Press any key to continue.
```

In the IO test we put a 100 words Cow essay. The application found two misspelled words "abhisek" and "Pooja".

We use the skip command "\s" to skip the first replacement and for the second case, we wrote "Puja". Now we can see the system replaced the "Pooja" with "puja" in the beginning of the third line.

Conclusion

Tiny spell checker is a lightweight spell checker application but it has all essential features that a spell checker need. This program can be used to find correction of a pre written letter, email or blog. It contains a huge amount of English words which will give you the correct output most of the time.

References

Index	Topic	Domain
1	SQLite C/C++ Documentation	SQLite
2	Stack Implementation Help	Stack
3	C++ Template Implementation	Tutorials Point
4	Reinterpret Cast and Other Helps	Stack Overflow
5	DB Browser for SQLite (Software)	SQLite Browser
6	Rlutil Color Changer Library	Rlutil
7	Code Blocks IDE	Code Blocks
8	C++ String Documentation	C++ String