|  |  |  |
| --- | --- | --- |
| **LAB101Assignment** | **Type:** | **LongAssignment** |
| **Code:** | **C.L.P0001** |
| **LOC:** | **150** |
| **Slot(s):** | **3** |

**Title**

Spell Checker.

**Background Context** bối cảnh

Many of us have horrible spelling and would get great practical use out of a spell-checker. In this assignment, you will write a simplified version of a **spell checker kiểm tra lỗi chính tả**. In the process, you will write several of your own string functions.

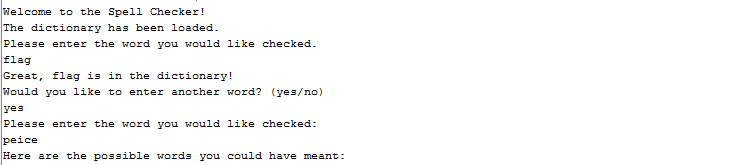
**Program Specifications** thông số

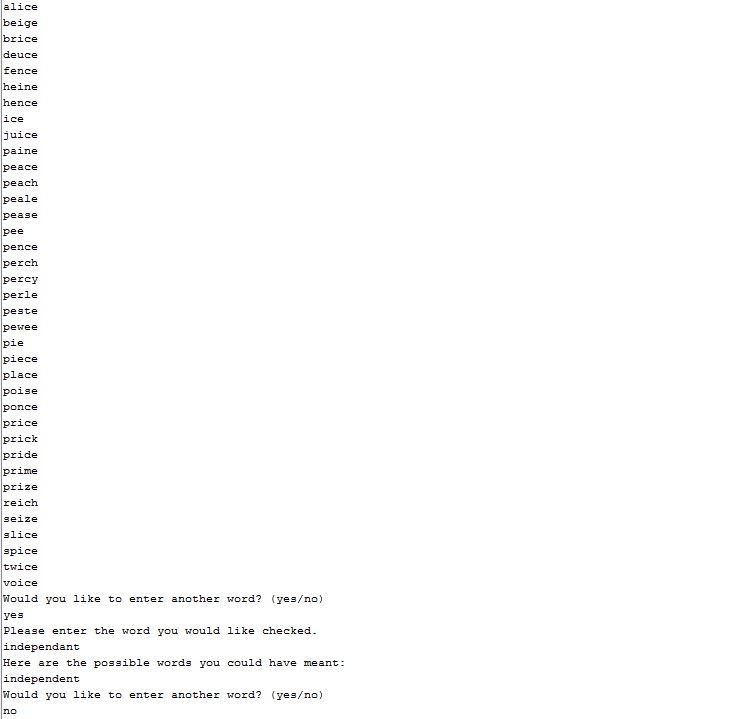
The first line of the file will have a single **positive integer** số nguyên dương, n (n ≤ 30000), **representing** đại diện the number of words in the file. The following n lines will contain one word each, in alphabetic order. All words will **contain lowercase letters chứa ký tự thường** only and will be no longer than 29 letters long.

***Function details:***

1. Your program should first read in the dictionary into a two-dimensional character array (from dictionary.txt) and print a message to the screen stating that the dictionary has been loaded.
2. Then your program should prompt the user to enter a word, all lowercase.
3. If the word is in the dictionary, you should simply print out a message stating this.
4. If the word is NOT in the dictionary, then you should provide a list of possible suggestions (of correctly spelled words that the user might have been trying to spell) in alphabetical order.
5. Here are the criteria for whether or not a real word should end up on this list:  
   Dưới đây là các tiêu chí cho việc một từ thực sự có nên kết thúc trong danh sách này hay không:
   * If either string is a substring of the other, and the lengths of the two strings are within 2 of one another.  
     Nếu một trong hai chuỗi là một chuỗi con của chuỗi kia và độ dài của hai chuỗi nằm trong phạm vi 2 của nhau.
   * If either string is a subsequence of the other and the lengths of the two strings are within 2 of one another. (This would get rid of the “car”, “camera” case, for example.)
   * If the strings are permutations of one the other. (Only try this test if the two strings are of the same length.)
   * If the matchscore between the two strings is less than 3. (Only try this test if the two strings are of the same length.)
6. Continue prompting the user to enter words until they decide that they don’t want to enter any more. (Thus, they will be forced to enter the first word, but after that, you’ll provide them a choice as to whether or not they want to enter another word to check

***Expectation of User interface:***





**Guidelines**

**Note: Assume that all of the formal parameters for each of the functions below are lowercase alphabetic strings.**

A string is a substring of another one if all of its letters appear contiguously (in the same order) in the second string. For example, “bound” is a substring of “homebound” and “ire” is a substring of “firefly”, but “car” is NOT a substring of “camera” because the letters ‘m’ and ‘e’ are in between the letters ‘a’ and ‘r’ in “camera.”

*// Returns true if shortstr is a substring of longstr,*

*// and false otherwise.*

*int substring(char shortstr[], char longstr[]);*

A string is a subsequence of another string if all the letters in the first string appear in the second string, in the same ordering, but not necessarily contiguously. For example, “car” is a subsequence of “camera”, since the letters ‘c’,’a’, and ‘r’ appear in that order (but not contiguously) in “camera.” Similarly, “hikes” is a subsequence of “chickens,” but “tale” is NOT a subsequence of “wallet” because the letter ‘t’ occurs AFTER the letter ‘a’ in “wallet.”

*// Returns true if shortstr is a subsequence of longstr,*

*// and false otherwise.*

*int subsequence(char shortstr[], char longstr[]);*

A permutation of letters is simply a different ordering of those letters. For example, “care” is a permutation of “race” and “spill” is a permutation of “pills,” but “apple” is NOT a permutation of “pale” because the letter ‘p’ appears only once instead of twice in “pale.” A natural consequence of this definition is that two strings can only be permutations of one another if they are both the same length.

*// Returns true if string1 and string2 are permutatations*

*// of each other, false otherwise.*

*int permutation(char string1[], char string2[]);*

When we are comparing two strings of equal length, we can define a term called “match score” which is simply the number of corresponding letters in which the two strings disagree. For example, the match score between “drink” and “blank” is 3 because the first three letters don’t match. The match score between “marker” and “master” is two because letters 3 (r vs. s) and 4 (k vs. t) do not match.

*// Precondition: string1 and string2 are the same length*

*// and only contain lowercase letters.*

*// Postcondition: returns the score of the match score*

*// between these two strings.*

*intmatchscore(char string1[],char string2[]);*

LAB101Assocation Loại: LongAssocation

Mã số: C.L.P0001

LỘC: 150

Khe cắm: 3

Tiêu đề

Công cụ kiểm tra chính tả.

Bối cảnh nền

Nhiều người trong chúng ta có chính tả khủng khiếp và sẽ sử dụng thực tế tuyệt vời từ một trình kiểm tra chính tả. Trong bài tập này, bạn sẽ viết một phiên bản đơn giản hóa của trình kiểm tra chính tả. Trong quá trình này, bạn sẽ viết một số hàm chuỗi của riêng bạn.

Thông số chương trình

Dòng đầu tiên của tệp sẽ có một số nguyên dương duy nhất, n (n ≤ 30000), đại diện cho số lượng từ trong tệp. N dòng sau sẽ chứa một từ mỗi từ, theo thứ tự chữ cái. Tất cả các từ sẽ chỉ chứa các chữ cái viết thường và sẽ không dài hơn 29 chữ cái.

Chi tiết chức năng:

1. Trước tiên, chương trình của bạn nên đọc từ điển thành một mảng ký tự hai chiều (từ dictionary.txt) và in một thông báo ra màn hình cho biết từ điển đã được tải.

2. Sau đó, chương trình của bạn sẽ nhắc người dùng nhập một từ, tất cả chữ thường.

3. Nếu từ đó có trong từ điển, bạn chỉ cần in ra một thông báo cho biết điều này.

4. Nếu từ KHÔNG có trong từ điển, thì bạn nên cung cấp một danh sách các gợi ý có thể (của các từ được đánh vần chính xác mà người dùng có thể đã cố đánh vần) theo thứ tự bảng chữ cái.

5. Dưới đây là các tiêu chí cho việc một từ thực sự có nên kết thúc trong danh sách này hay không:

o Nếu một trong hai chuỗi là một chuỗi con của chuỗi kia và độ dài của hai chuỗi nằm trong phạm vi 2 của nhau.

o Nếu một trong hai chuỗi là một chuỗi con khác và độ dài của hai chuỗi nằm trong phạm vi 2 của nhau. (Ví dụ, việc này sẽ loại bỏ trường hợp xe hơi, ví dụ về máy ảnh, máy ảnh.

o Nếu các chuỗi là hoán vị của nhau. (Chỉ thử kiểm tra này nếu hai chuỗi có cùng độ dài.)

o Nếu điểm trùng khớp giữa hai chuỗi nhỏ hơn 3. (Chỉ thử kiểm tra này nếu hai chuỗi có cùng độ dài.)

6. Tiếp tục nhắc người dùng nhập từ cho đến khi họ quyết định rằng họ không muốn nhập thêm nữa. (Do đó, họ sẽ bị buộc phải nhập từ đầu tiên, nhưng sau đó, bạn sẽ cung cấp cho họ một lựa chọn về việc họ có muốn nhập từ khác để kiểm tra hay không

Mong đợi của giao diện người dùng:

Hướng dẫn

Lưu ý: Giả sử rằng tất cả các tham số chính thức cho mỗi hàm bên dưới là các chuỗi chữ cái viết thường.

Chuỗi là một chuỗi con của một chuỗi khác nếu tất cả các chữ cái của nó xuất hiện liền kề (theo cùng một thứ tự) trong chuỗi thứ hai. Ví dụ: Giới hạn về mối quan hệ là một chuỗi con của hom homoundound và tính năng là một chuỗi con của con đom đóm, nhưng chú chó không phải là một chuỗi con của máy ảnh bởi vì các chữ cái 'm' và 'e' nằm ở giữa các chữ cái 'a' và 'r' trong máy ảnh.

// Trả về true nếu shortstr là chuỗi con của longstr,

// và sai khác.

chuỗi con int (char shortstr [], char longstr []);

Một chuỗi là một chuỗi của một chuỗi khác nếu tất cả các chữ cái trong chuỗi đầu tiên xuất hiện trong chuỗi thứ hai, theo cùng một thứ tự, nhưng không nhất thiết phải liên tục. Ví dụ, trên mạng, máy tính là một phần tiếp theo của máy ảnh, vì các chữ cái 'c', 'a' và 'r' xuất hiện theo thứ tự đó (nhưng không liền kề) trong máy ảnh. sự xuất hiện của những con gà con, những người khác, nhưng những câu chuyện về thế giới không phải là sự xuất hiện của ví điện tử vì chữ 't' xảy ra SAU chữ cái 'a' trong ví điện tử.

// Trả về true nếu shortstr là một chuỗi con của longstr,

// và sai khác.

int sau đó (char shortstr [], char longstr []);

Một hoán vị của các chữ cái chỉ đơn giản là một thứ tự khác nhau của các chữ cái. Ví dụ, Chăm sóc sức khỏe là một hoán vị của cuộc đua và sự cố tràn lan là một sự hoán vị của những viên thuốc, nhưng nhưng táo táo không phải là một hoán vị của Truyện nhạt vì chữ 'p' chỉ xuất hiện một lần thay vì hai lần trong Càng nhạt. Một hệ quả tự nhiên của định nghĩa này là hai chuỗi chỉ có thể là hoán vị của nhau nếu cả hai cùng độ dài.

// Trả về true nếu string1 và string2 là hoán vị

// của nhau, sai khác.

int hoán vị (char chuỗi1 [], char chuỗi2 []);

Khi chúng ta so sánh hai chuỗi có độ dài bằng nhau, chúng ta có thể định nghĩa một thuật ngữ gọi là điểm số phù hợp với điểm số đơn giản là số lượng chữ cái tương ứng trong đó hai chuỗi không đồng ý. Ví dụ: điểm số trận đấu giữa các thức uống của người Bỉ và trò chơi trống trống là 3 vì ba chữ cái đầu tiên không phù hợp. Điểm số trận đấu giữa các điểm đánh dấu của người Hồi giáo và người làm chủ cuộc trò chuyện là hai vì các chữ cái 3 (r so với s) và 4 (k so với t) không khớp nhau.

// Điều kiện tiên quyết: string1 và string2 có cùng độ dài

// và chỉ chứa các chữ cái viết thường.

// Postcondition: trả về số điểm của trận đấu

// giữa hai chuỗi này.

intmatchscore (char chuỗi1 [], char chuỗi2 []);