CPP Problem Design Example

Subject: Levenshtein Distance

Contributor: 溫勇威, 陳靖升, 鍾賢廣

Main testing concept: Array and String

Basics

■ C++ BASICS
■ FLOW OF CONTROL
■ FUNCTION BASICS
□ PARAMETERS AND OVERLOADING
■ ARRAYS
□ STRUCTURES AND CLASSES
□ CONSTRUCTORS AND OTHER TOOLS
□ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES
■ STRINGS
■ POINTERS AND DYNAMIC ARRAYS

Functions

□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
□ RECURSION
□ INHERITANCE
□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ TEMPLATES
□ LINKED DATA STRUCTURES
□ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

Description:

Levenshtein distance is a measurement of the degree of similarity between two words. Levenshtein is the minimum number of steps consumed by the operation of inserting, deleting, or replacing one word into another word.

For example, the Levenshtein distance between "kiitten" and "sitting" is 4. There is no other way to make the switch with fewer steps.

(1) kiitten -> siitten (substitution of “s” for “k”)
(2) siitten -> sitten (deletions of “i” at the third place of siitten)
(3) sitten -> sittin (substitution of “i” for “e”)
(4) sittin -> sitting (insertion of “g” at the end)

Note: Upper letter and lower letter are considered different letters.

Input:

A series of words, each two words is a set. Find the Levenshtein distance of them.

Output:

Find the minimum distance between two text and print that number(int).

Sample Input / Output：

Sample Input	Sample Output
Google Facebook Winter is coming Here comes Winter I am the bone of my sword. Steel is my body and fire is my blood. I have created over a thousand blades. Unknown to death. Nor known to life. I am the bone of my code. Steel is my structure, and fire is my algorithm. I have fixed over a thousand bugs. Unknown to dawn. Nor known to night.	8 14 37

■ **Easy, only basic programming syntax and structure are required.**
□ **Medium, multiple programming grammars and structures are required.**
□ **Hard, need to use multiple program structures or complex data types.**

Expected solving time: 30 minutes
Other notes: