**CPP Problem Design Example**

|  |
| --- |
| **Subject: Levenshtein Distance** |
| **Contributor: 溫勇威, 陳靖升, 鍾賢廣** |
| **Main testing concept: Array and String**   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ 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 | □ 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 method of similar strings which measuring the difference between two sequences. The Levenshtein distance between two words is the minimum number of single-character edits.(insertions, deletions or substitutions)  For example, the Levenshtein distance between "kiitten" and "sitting" is 4. There is no way to do it fewer than four edit.   1. **k**iitten -> **s**iitten (substitution of “s” for “k”) 2. siitten -> sitten (deletions of “i” at the third place of siitten) 3. sitt**e**n -> sitt**i**n (substitution of “i” for “e”) 4. sittin -> sittin**g** (insertion of “g” at the end)   Note: Upper letter and lower letter are considered different letter.  **Input:**  Enter two paragraphs of text and separate by Enter. This program allows multiple case. User can enter until read EOF.  Note: Input must be in the range of ASCII.  **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 | |
| **■ Eazy,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:** |