Data Structure Project 1
Keyword Search

簡介

實作 string matching 的演算法，KMP(Knuth-Morris-Pratt)與 Brute-Force，找出 keyword 在文件裡的全部位置，方式實作，並比較執行時間。

- 請勿使用 C++ STL，所有演算法需自行實作
- 請勿使用任何與字串相關的 library，若不知道是否能夠使用，請寄信詢問助教
- 字串請自行以 array 實作，請勿使用 C 與 C++中的 string
- 以 C/C++開發，可接受 VC2008 以上, GCC4.6.3 以上

輸入輸出格式要求

- 輸入：YOUR_PROGRAM PATTERN_FILE TEXT_FILE
  - YOUR_PROGRAM: 程式執行檔
  - PATTERN_FILE: 要搜尋的 pattern(純文字檔)，每個 pattern 一行，pattern 本身不含換行字元，一個 pattern file 中可能有多個 pattern
  - TEXT_FILE: 被搜尋的純文字檔
  - Pattern file 與 text file 檔案內容皆為 ASCII，不含中文，單行最多 2000 個字元， 行數不限。
  - 程式從命令列讀取兩個參數，第一個參數為 pattern file，第二個參數為 text file，例如你寫的程式執行檔為 project1.exe，pattern file 是 1.txt，text file 是 a.txt，則執行命令應為「project1.exe 1.txt a.txt」，程式 project1.exe 會讀取 1.txt 與 a.txt，並且搜尋與 pattern 相同的字串。
- 輸出：LINE_NUMBER OFFSET PATTERN STRING
  - LINE_NUMBER: pattern 在 text file 中的第幾行被搜尋到
  - OFFSET: pattern 在搜尋到的該行中，從第幾個字元開始
  - PATTERN: 搜尋的 pattern，請在 pattern 左右加上雙引號，如”abc”
  - STRING: 搜尋到的該行，請在 string 左右加上雙引號，如”I am hungry.”
  - 英文不區分大小寫
  - 各項之間使用空白隔開
  - 輸出所有搜尋到的字串後，請輸出計算時間(以 ms 為單位計算)

範例 1

1.pat (PATTERN_FILE)	mile
1.txt (TEXT_FILE)	you smile, I smile and everyone smiles
程式命令參數	1.txt a.txt
程式輸出	1 6 “mile” “you smile, I smile and everyone smiles” 1 15 “mile” “you smile, I smile and everyone smiles” 1 34 “mile” “you smile, I smile and everyone smiles” 0.001ms

範例 2

2.pat (PATTERN_FILE)	Xx oo
2.txt (TEXT_FILE)	xxxxx ooxx
程式命令參數	2.txt b.txt
程式輸出	1 1 “Xx” “xxxxx” 1 2 “Xx” “xxxxx” 1 3 “Xx” “xxxxx” 1 4 “Xx” “xxxxx” 2 1 “oo” “ooxx” 2 3 “Xx” “ooxx” 0.001ms

範例 3

3.pat (PATTERN_FILE)	123a b
3.txt (TEXT_FILE)	3123a bpp 12 3ab Qq13fB
程式命令參數	3.txt c.txt
程式輸出	1 2 “123a b” “3123a bpp 12 3ab” 0.003ms

作業繳交內容

- 程式
  - 請繳交程式碼與一個測試檔
  - 程式執行時需搭配參數，例如「program.exe 1.txt a.txt」
  - 程式結果請輸出到標準輸出(stdout)。
  - 測試檔的執行將結果紀錄在報告中，並簡單說明內容
  - 程式碼可讀性為評分標準之一，包含註解、變數名稱、函數名稱等
  - 若使用 VC++，請編譯為 release 版本，並將整個 project 上傳
  - 若使用 GCC，請確認執行檔能在工作站上執行
- 報告
  - 請繳交 pdf 格式，檔名取為學號_proj1.pdf，例如：9876543_proj1.pdf
  - 實作演算法介紹
  - 需包含測試資料與結果並說明
  - 需比較 Brute-Force 與 KMP 的執行時間
  - 其他特色與說明

評分標準

- 程式與程式碼 40%
  - 程式碼正確性、可讀性與註解 20%
  - 執行速度 20%
- 報告 30% (包含自我測試資料與時間比較)
- 助教測試資料 40%，助教將自行測試數份資料(30%基本分，10%加分題)，執行正確即可獲得分數
- DEMO 將會隨機抽取學生進行測試、講解