

程式設計大賽於 2004 年起第一次舉辦至今，已為 Openfind 年度重要的盛事及傳統之一。公司內部人員於比賽期間將打破原部門建置，重新編組，依據比賽題目內容，並在有限的時間及資源下，發揮最大的創意及團隊合作，努力達成目標。

2005 年 Openfind 程式設計大賽花絮如下：

題目：Compression. 比賽時間：24 小時

競賽規則：

- (1) 每項目表現最佳得 10 分、第二名 9 分、第三名 8 分，依此類推
- (2) 壓縮比例佔 70%，壓縮/解壓縮速度佔 30%
- (3) 壓縮/解壓縮失敗、程式錯誤等，該項目零分
- (4) 總分最高者冠軍、第二亞軍、第三季軍

比賽結果名次公布：

- 第一名：**Howard, Peggy, McLee**
- 第二名：**Sylvia_Chen, Stewart, Jason**
- 第三名：**Christopher, Chinting, Sorry**

測試檔案內容：

Input filename	Original type	Description
Sample1.wav	50Mb Wav 檔案	Chopin 鋼琴曲
Sample2.wav	23MB EXE 檔案	StarCraft 程式
Sample3.dat	31MB Wav 檔案	Phantom of the Opera
Sample4.doc	20MB Word 檔案	某系統管理手冊
Sample5.mp3	3MB MP3 檔案	Mozart 鋼琴曲
Sample6.exe	8MB EXE 檔案	Word 執行檔
Sample7.txt	7MB MP3 檔案	藍色多瑙河
Sample8.txt	11MB TXT 檔案	Office XP Log file
Sample9.exe	1MB TXT 檔案	古龍小說一篇
Sample10.exe	2MB DOC 檔案	請假系統操作手冊

結果 - 壓縮比例分數佔 70%，花費時間比例佔 30%：

Team	1	2	3	4	5	6	7	8	9	10	Result
五	壓縮比: 8 分	壓縮比: 9 分	壓縮比: 8 分	壓縮比: 7 分	壓縮比: 6 分	壓縮比: 7 分	壓縮比: 7 分	壓縮比: 8.5 分	壓縮比: 7 分	壓縮比: 6 分	69.15 分
	速度: 5 分	速度: 3 分	速度: 5 分	速度: 7 分	速度: 6 分	速度: 8 分	速度: 6 分	速度: 4 分	速度: 9 分	速度: 6 分	
三	壓縮比: 7 分	壓縮比: 9 分	壓縮比: 5 分	壓縮比: 7 分	壓縮比: 6 分	壓縮比: 7 分	壓縮比: 7 分	壓縮比: 8.5 分	壓縮比: 7 分	壓縮比: 6 分	66.2 分
	速度: 6 分	速度: 4 分	速度: 6 分	速度: 6 分	速度: 7 分	速度: 7 分	速度: 5 分	速度: 3 分	速度: 7.5 分	速度: 7 分	
九	壓縮比: 10 分	壓縮比: 9 分	壓縮比: 0 分	壓縮比: 7 分	壓縮比: 6 分	壓縮比: 7 分	壓縮比: 7 分	壓縮比: 8.5 分	壓縮比: 7 分	壓縮比: 6 分	60.9 分
	速度: 10 分	速度: 1 分	速度: 0 分	速度: 5 分	速度: 5 分	速度: 6 分	速度: 4 分	速度: 2 分	速度: 7.5 分	速度: 5 分	
七	壓縮比: 6 分	壓縮比: 6 分	壓縮比: 4 分	壓縮比: 4 分	壓縮比: 4 分	壓縮比: 5 分	壓縮比: 5 分	壓縮比: 5 分	壓縮比: 4 分	壓縮比: 4 分	59 分
	速度: 9 分	速度: 10 分	速度: 10 分	速度: 10 分	速度: 9 分	速度: 10 分	速度: 7 分	速度: 9 分	速度: 4 分	速度: 9 分	
四	壓縮比: 0 分	壓縮比: 1.5 分	壓縮比: 6.5 分	壓縮比: 7.5 分	壓縮比: 9.5 分	壓縮比: 9.5 分	壓縮比: 9.5 分	壓縮比: 2.5 分	壓縮比: 9.5 分	壓縮比: 9.5 分	58.95 分
	速度: 0 分	速度: 8 分	速度: 3 分	速度: 3 分	速度: 4 分	速度: 4 分	速度: 3 分	速度: 5 分	速度: 5 分	速度: 4 分	
六	壓縮比: 0 分	壓縮比: 1.5 分	壓縮比: 6.5 分	壓縮比: 7.5 分	壓縮比: 9.5 分	壓縮比: 9.5 分	壓縮比: 9.5 分	壓縮比: 2.5 分	壓縮比: 9.5 分	壓縮比: 9.5 分	58.05 分
	速度: 0 分	速度: 9 分	速度: 2 分	速度: 2 分	速度: 3 分	速度: 3 分	速度: 2 分	速度: 10 分	速度: 3 分	速度: 2 分	
八	壓縮比: 5 分	壓縮比: 5 分	壓縮比: 3 分	壓縮比: 5 分	壓縮比: 3 分	壓縮比: 4 分	壓縮比: 4 分	壓縮比: 6 分	壓縮比: 3 分	壓縮比: 3 分	55.1 分
	速度: 8 分	速度: 7 分	速度: 9 分	速度: 9 分	速度: 10 分	速度: 9 分	速度: 8 分	速度: 8 分	速度: 10 分	速度: 10 分	
一	壓縮比: 4 分	壓縮比: 4 分	壓縮比: 10 分	壓縮比: 3 分	壓縮比: 8 分	壓縮比: 3 分	壓縮比: 3 分	壓縮比: 1 分	壓縮比: 2 分	壓縮比: 8 分	46.6 分
	速度: 3 分	速度: 5 分	速度: 7 分	速度: 4 分	速度: 2 分	速度: 5 分	速度: 10 分	速度: 7 分	速度: 2 分	速度: 3 分	
二	壓縮比: 9 分	壓縮比: 7 分	壓縮比: 9 分	壓縮比: 0 分	壓縮比: 0 分	壓縮比: 0 分	壓縮比: 0 分	壓縮比: 8.5 分	壓縮比: 5 分	壓縮比: 0 分	32.05 分
	速度: 4 分	速度: 2 分	速度: 4 分	速度: 0 分	速度: 0 分	速度: 0 分	速度: 0 分	速度: 1 分	速度: 6 分	速度: 0 分	
十	壓縮比: 3 分	壓縮比: 3 分	壓縮比: 2 分	壓縮比: 2 分	壓縮比: 2 分	壓縮比: 0 分	壓縮比: 2 分	壓縮比: 4 分	壓縮比: 0 分	壓縮比: 2 分	32 分
	速度: 7 分	速度: 6 分	速度: 8 分	速度: 8 分	速度: 8 分	速度: 0 分	速度: 9 分	速度: 6 分	速度: 0 分	速度: 8 分	

幕後花絮

Keynes

在我的心中我很期望有一天能和黃柏雄在程式面一決勝負，他真的超能實作，我遠遠比不上，不過我仍期望有一天能和他對決，能和他同一組是超好的經驗，他無法上台北，我想了想，要獲勝的方法就我和 david 幫他把程式控管好（因為遠端二人一起工作的品質很難管控，而現況而言他的實作速度遠比我快，但需要二個人幫忙確認程式正確性及討論程式方向〔什麼該作而什麼不該作，要作什麼才會贏〕），基本上往這方向想是成功的也是合理的，可是最後我仍有些事忘記了，我和他太想贏，最後一分鐘還在改新方法，至少要應該留一小時作全面的測試，有作全面程式的測試才是獲勝的最後重點，我覺得比賽就是要在規則下努力去贏，我很想和黃柏雄在程式下決一勝負，但在同一組下就是要想贏的最有可能方式，在他沒上來的情況下我要作什麼角色才是合適的，我覺得人作事的角色是多變的，今天可能是 RD 明天可能是 SE，也許後天會去做業務，PM，重點是如何扮演好這個角色，這個角色需要什麼，這個角色不需要什麼？我覺得這個比賽很有趣一開始可以把自己的角色定位好，協同合作，這個很像玩 13 張，要看手上有什麼牌來決定要怎麼定出要怎麼比賽，這真的是很有趣的事，這次是一個好教訓，看到前 3 個 test case，黃柏雄的程式是第一名，但因為選擇用最後的方法(沒有時間測的方法)，卻影響最後的結果，真感到悲情，不過反過說我很慶幸用沒測過的最後程式是失敗了，因為這樣子我才會記住出貨的東西 always 是要被驗證的，有限的時間，一定有些東西是賭一把，但有些東西是不能賭的，這些事是牢記的，總之我覺得公司程式比賽迷人的地方在於以我們 team 的資源要如何在限的時間獲勝，我這次要作什麼角色才是最有利的打算，然後剩下的就是忠實及努力的執行自己的規劃了

另外有一件好笑的事本來我週六是要出去玩的，我也橋好我週五和黃柏雄作一天的合作，然後週六，早上 9:30 左右離開，但人算不如天算，沒想到他上不來，只能 remote 比賽，這個打亂了我全部的布局，我變的勢必要比到 12:00，而他上不來我要怎麼和他合作，這個變局是很有趣的（也很無奈的），在突來的變數下決定自己的角色是一件有趣的事，因為重點不是愛作什麼事，而是要作什麼事，不管愛不愛都努力作好它，這真是一個有趣的經驗:)

C_Liao 我們選用的外部 library 是 zlib 和 monkey audio, 但最後上場的只有 zlib. 原因是雖然 monkey audio lib 在 survey 及 prototyle 的階段跑起來都很順, 無奈我們到最後才發現解檔時它只認自己的 .ape 的附檔名, 那時已經沒剩下多少時間改, 所以只能用 zlib 上場.

失敗的地方;

1. 應該早點測出 monkey audio 的問題, 加以解決
2. 浪費太多時間在 survey 及整合其它 lib, , 而應該花更多時間在測試及調整主要的 zlib.

	總結的來說： 1. 針對需求的, 正確的 survey 很重要. 2. open source 的軟體可能好用, 但是 open source 的 library 可能就沒那麼好用, 特別對 win32 平台可能有更多限制, 所以要額外小心. 3. 連續長時間 coding 下來腦子會累！眼睛會花！心情會鬱卒！
John	來到公司一個月，就有機會參加年度程式設計比賽，對我而言是意外的驚喜。記得上回參加程式設計比賽， 是就讀高中三年級，當時對資料結構和演算法完全不了解，一遇到複雜的題目就全掛點了。 這次和同事廖拯、秋賢一起為比賽而努力，雖然最後沒有得名，對我而言卻是一次相當寶貴的經驗。 因為一個程式能成功，除了傑出的演算法，還需要穩健的整合和除錯，這是在賽前無法想像的。 很感謝公司給我們這個機會參與，也希望能於明年的活動中締造佳績！
Bryson	看大家為了程式設計比賽熬夜打拼真是感動， 雖然我是打雜的參與的不多，但還是很感謝廖拯和 john 的辛苦付出了。
Ferro	當然是緊張又刺激 第一種刺激, 明顯得讓人要有"陪榜"準備. 一些人動作快得讓人猜不透呀! 我在"先吃中飯"還是先想"題目"時, 好像有其他組別,分工已經清楚了; 自己還在茫茫網海找資料時, 人家已經在分析那一種演算法對付那一種檔案; 好不容易動手寫程式了, 他們已經在測試了. 第二種刺激 感受到,"前輩"經驗老到,和實力堅強 coding 熟不熟, 一下子就被測出來了. 不少基本的 coding,我還是必須查查範例,翻翻文件. 浪費不少寶貴的時間. 求穩和不當機,已花去我了大部分的時間, 來自於我的助力還是有限. 多半得感謝"前輩們"的指導和修正 壓力和助力下, 我們還是完成我們想做的 雖然,最後沒成為"最好的" 不過,如同 Felka 所說,(和 Felka,Daniel 同組) "先求有,再求好"
Chinting -	題目一公佈，

1	第一組立即開始動作， A: 你們去找資料，程式交給我，我馬上就動工.. 時間一分一秒地過去了..... 我們組的吃完飯回來... 看到 A 已經在 coding... 我說：哇！幹麼...太拼了吧你們，都不去吃飯的唷.... A：我才要問你在幹麼咧～你們組的還有時間去吃飯喔.. 我臉上好幾條線 -_- " 乖乖地坐下來找資料... 過沒多久， 就聽到 A 拿起電話來： "我現在給你第一版！你可以開始測了！測完跟我講結果.." 我在心裡想："氣...坐你旁邊壓力太大了..." 再隔著半小時左右， 又聽到 A 拿起電話來："給你第二版的！你再測看看，等下會再給你最新版！" 這時候的我，已經在心裡什麼話都想罵了："是怎樣～過份！過份～～" 後來才知道， peggy 跟我一樣，當時是既緊張又生氣~~~~~ 因為 A 就是在跟 peggy 前面的 Tony 連線...-_- 嗯， 第一組其實是最快寫完，又最早有完整版出來的人， 今天比賽時間如果只有 6 個小時，甚至是 3 個小時， 他們絕對是贏定了..
Chinting - 2	換第二組測試了， 老 K： 我好緊張，我只要求它不要當就好... 某人回說：你怎麼可以這樣呢...要有信心 Y.... 老 K：我也想阿！但是我們大師兄只留給我 1 分鐘測試， 我還測到有當掉的狀況，我當然擔心阿...
Chinting - 3	輪到我們了~~輪到我們了~~~ 第九組，第九組.. 快，快，compile， bala..bala..bala..十二，二十五，三十，五十六，八十，八十二， 大家一片讚嘆聲："哇！82 個 Warning .." 我們組員異口同聲地說："放心，放心，Warning 而已！程式是可以跑的，而且是對的啦！" 隔了 3 秒鐘， Sorry 遲疑地說了句："咦，比剛剛少了 1 個 Warning 耶...怎辦...我們有 Copy 對的版本下來嗎？" hmmm....別鬧了....-_-"

Chinting - 4	蛋哥：出這什麼爛題目阿，一點都不有趣... 大家看著出題者 - phantom，然後發出一陣驚呼聲，(喔~~~厲害喔...蛋哥..) 這時 phantom 臉上應該冒出好幾條線 -_-" 然後咱們偉大的蛋哥又接著說了： "本來就是阿，怎不出個遊戲呢，還比較有趣一點..." 終於， 某 H 替 phantom 發出正義之聲： "出一題遊戲給他吧!....然後....他寫就好....明天交...." 大家立即齊聲說好~~~~ 某 C 也說話了："那明年給他出個象棋好了..." 停頓了 3 秒鐘..... 某 H 回說："好阿，蛋哥比象棋，我們其它人比西洋棋...." 這時，大夥真是開始要起立鼓掌通過了~~~ 氣氛 high 到不行.... 後來，更是此起彼落地每人接一句， "別這樣，蛋哥等下真的就回去開始寫象棋，為明年做準備了耶.." hmmm....我不懂...為什麼蛋哥的一句話...讓大家開始一人一句地接了起來呢?? 果然，前輩就是不一樣啦.....-_-"
Chinting - 5	評分開始， phantom 找人要把程式跑的結果記在白板上， 底下大家開始你一句我一句地討論起來， "用 Excel 就可以匯入，再統計分析就好了 Y..." "快！誰誰誰...馬上寫個 parser 出來..." "快！那個 AWK 不是很好用嗎...." 等到計分完畢，面對著滿滿的白板的一堆數字，開始要算總分了， 大家又開始你一句我一句地討論起來， 結果不曉得是誰，就蹦出了這麼一句話："有時間寫賭博網站，怎不先寫個計分程式出來？！?" 大家一陣狂笑， 此時，phantom 手上握著白板筆，停了下來，臉上早不知道有幾條線了.....-_-"" 他心底一定有個聲音在說： "我全都賭輸了，還要在這裡幫大家計分，我..我...我最可憐好嗎...." hmm....我相信...phantom 是用心良苦的... 他寫賭盤網站，一定是想讓沒有參加比賽的人，也有參與感...

	他沒有寫計分程式，一定是想要讓大家有慢慢等結果的刺激感... 是吧？！ 您說，是吧？！ (phantom，這樣有糖吃嗎?...呵.....)
Chinting - 6	比賽結束， 我們組開起了檢討會， 討論到最後，得到 1 個結論， 如果多加一行判斷， 也許就有機會可以拿第一了， 而那一行判斷， 平均下來， 1 個字母大概損失了\$1600.. (是 1 個字母，不是 1 個單字喔....) Sorry 感嘆地說："從沒有寫過這麼貴的 code 吧！ >_<"
Andric	今天來和大家講一個龜兔賽跑的故事... 當題目一公佈之後，兔子一號立刻就把工作都分下去：兩個人負責找壓縮的演算法，一個人負責準備測試資料。啪啪啪...一下子就把所找到的資料都先印出來，A 和 W 先生就來負責 implement 每一個有找 source code 的演算法，寫完一個，就丟一個給 T 先生測，biacode, bw_trans, bwt_zip, huffpack, kexis, lzw1, rushuf1 ... 等。 很快地，就把各種檔案型態的較佳壓縮演算法加以組合，那就是 bwt_zip 搭配 Kexis (專門對付 WAV)，那就是 ---- AWT (Andric, Wallace, Tony)。 這一切都是這麼美好，當 AWT 出來之後，才不過是禮拜五的晚餐時間，好像一切就已經都忙完了。所以就開始針對表現最差的 MP3 檔案來做處理，不過弄了一個晚上，卻沒有什麼好的結果。這時候還不知道大禍臨頭的 A 先生居然開始去改壓縮檔案的格式，希望可以在擠出多一點空間，來賺取效能的分數。 但是，一切變化就從禮拜六悠哉的早餐開始。「咦～怎麼 1 M 文字檔跑不完？怎麼 48 M 的 WAV 也會當？天啊~~~」離截止時間只剩下最後兩個小時了，沒有時間來解那兩個演算法的 bug，又沒有勇氣去賭程式在比賽的十個檔案裡是不會當。所以就開始來調不會當的配方。當檔案大小超過 xxx 的時候，採用方案 A。最後，結果很明顯，我們只有用到 bwt_zip 和 kexis 演算法的檔案表現比較好，至於其他的檔案都輸的很慘。真是對不起押第一組的同胞們~~~ 這個故事告訴我們，一開始做的比較快，並不一定真的會贏，而且大部分的情況，最後都不會有好結果。另外就是認真的小組最優秀了，前三名的隊伍都是留到一點多以後的喔，所以說...留得越晚，名次越好。
Howard	團隊合作才是致勝關鍵，每一個角色都一樣重要 在大家高興著賭盤開出如預期結果的時候，有幸成為冠軍隊裡一員的我，其實心裡還有一點心虛， 老實說，在比賽結果還未正式出爐之前，我覺得我們這一對並沒有什麼勝算。 雖然我們擁有 RD 之花 - 孟秋 -，但是我想大家也都有個先入為主的觀念，一定會覺得如果一個 team 裡 ，分到 3 個人都是 RD 的話，Coding 實力應該會比較堅強....而分組名單裡，我們 team 裡的 RD 是我跟孟秋。 偏偏，在比賽前我得了重感冒，比賽前一天的颱風假，大家在家裡休息的時候，我是在家

	裡 發燒加上吐下瀉，耗了一整天。 老實說，星期五當我還有考慮過要請病假在家休息一天，不過這樣擺明了就是叫孟秋自己看著辦， 大男人主義的我覺得這樣有欺負弱女子的嫌疑(雖然事後證明我應該是欺負不到她)，所以我當天 還是準時出現在公司。 星期五中午公佈題目時，我本來以為要來一場演算法大賽了，所以正打算跟孟秋好好討論戰術...不過仔細想想， 再加上偷偷打聽其他 team 的策略，我發現同事們大多是打算包現成的函式庫，而且遊戲規則似乎也沒有禁止這 種行為...由於現成的函式庫就那幾套，到最後大家寫出來的程式應該會都差不多，所以， 最後決勝的關鍵也許 就是哪一組能夠好好調校函式庫，發揮出最佳效能...我事後想想，這一點也許是我們能致勝的關鍵之一。 在我們開發程式的過程中，我們的另一位我一直都沒有提到的隊員 - Peggy - 所負責的工作是準備測試 sample 跟測試我們這組的程式。老實說，她在這次比賽過程中也佔了重要的角色，因為她準備了充分的 test 環境 及 sample，使我們可以專心的寫程式，而且程式是一寫完就有人可以幫忙做完一連串完整測試，甚至還會作成 Excel 報表，這點對於 debug 或是調校效能時幫助很大。 記得在開發過程中，有一次我們出了一個新版程式，我急著想知道它 Run 起來的結果如何，偏偏 Peggy 剛好去上廁所，心急的我自己就跑到她的測試環境先測起來了，這時候我真的深深感受到，QA 的工作也 不是每個人都作得來的，反覆而繁瑣的測試，還要注意很多因素.... Peggy 啊~妳真是有耐心的人。 當然孟秋依然是最重要的角色，因為她快速的先包出一套演算法，讓我們這一組可以有了測試的基準，而且 比賽結果證明，她的程式是大功臣。這次的比賽過程裡，我們這組的程式在每個測試項目裡大多不是最強的， 不過我們在每個項目都保持平均以上的成績，再加上很多組都在一些測試項目裡失敗了，所以最後我們拿到 了冠軍。要是讓我事後回想我們會致勝的關鍵，我想，是因為各位下注在我們身上的同事，他們想贏錢 的念力....還有還有孟秋高品質且穩定的演算法，再加上 Peggy 的努力，提升了我們程式的品質，讓我們的程式 穩定不會失敗，還能保持一定的水準。 我想有人一定很還懷疑，看了這麼多，那到底另一位 RD 隊員 - Howard - 比賽期間都在做些什麼事？老實說，這位隊員比賽期間作最多的事就是.... 不停的包餛飩..... Christopher！我承認！你桌上那一包衛生紙是被我包完的，別打我 --。
Williams	「無中生有的 code 往往會造成非預期的結果」 by 程式設計比賽第十名

Richard	1. 先找個 general purpose 的壓縮 library 2. 再找找看有無針對 wav 的 library 3. 預防檔案過大，無法於時間內完成，要判斷檔案大小，必要時不壓縮 上述策略也 OK，然而實作上來不及整 wav 壓縮，而採用的 zlib，速度快，但是壓縮比不高，使得分數的總合較低。 回頭來看，在時間的限制下，實作是很重要的，不必太貪心，先求一台可以出賽的車子，有時間再去細部校調。
Lungchen	我們從題目宣佈以來，一直進度緩慢，所以決定策略是只要穩定就好，只要程式能正確壓縮、解壓縮，而且能在一分鐘內跑完，不要有項目掛蛋就好，至少在速度上拿分，也許還能超過一些壓縮慢到超過一分鐘的組。 所以拿了 zlib 來簡單套一下，沒做什麼大變動，就上場了。 結果，有達到預期的效果，速度項目都拿了 8,9,10 分，可是只在速度的三成分數搶分，沒把握到更重要七成的壓縮比，所以成績不高。沒關係，明年再來，下次會更把握重點，yeah。
Solos	很榮幸第二次參加比賽，雖然還是沒有幫到忙，但是可以學到在競爭之下，必需調整到對自己最有利的方 式(例如應該著重在壓縮比) 而且這次比賽，也看出 RDs 其實都蠻團結的，賭博/加油 網站也增進了不少公司的向心力。 另外個人最大的收穫是，我居然在這兩天把給 bryson 的線上交接文件寫完了..
Joyen	雖然我們這組並沒有得名，但是並不覺得遺憾 因為原本不熟悉的三個人，透過這次的比賽，知道彼此的存在 在互相體諒與包容的氣氛下，把一個作品完成 雖然學生時代參加過不少比賽，但是那些比賽，往往一比，短則三天或是一個星期，甚至也參加過為期一年的比賽， 但是參加過兩次公司舉辦的比賽，每當比賽結束時，總有一種意猶未盡的感覺 很希望下次的比賽，可以讓我們關在公司一個禮拜，讓我們可以一輩子都難以忘懷
McLee	先說結論，我覺得 QA 真的很重要。有了 QA，第一是可以專心做東西，這真的很棒；第二更重要， 就是你做出來、交出去的東西，自己和別人都會很安心；而且是超過 100%的安心感。 這結論是之前的我只能想像未曾體會的感 覺，之前寫程式，包括上次程設比賽，一向都是自己測自己的東西 (或說 RD 測自己或別人，但也是 RD 寫的東西)，自己做 unit test 是一個 programmer 最基本的負責表現， 所有不同狀況、回傳值等等，自己都應該要基本測過；不過這次有了 Peggy，我才見識到什麼叫測試! 呵! 那種感覺真的是.. 只能說用她的偉大的耐心換取大家的安心~ 所有能測的、不能測的狀況都給她跑過了， 我們一般做的 unit test 真的只能說是盡了最小的責任.. 哈哈 :D 有 QA 的感覺，真的很好!! 另外的心得應該就要說是連莊心得吧! 上次比賽我承認我們這組真的花了比別人多的心力..

	雖然核心也不是說多強 :p,但是整個系統堪稱在有限時間內,儘量完整地實作了每個層面。這次的題目不同,不是做系統,而是做一件很單純的演算;我一看到題目,立刻改了 msn 暱稱, 因為不希望大家把太多賭金寄在我身上 ^^; 因為我覺得這不是我擅長的東西...真的怕有負眾望。 所以說,為何最後會連莊,我想真的,實力的層面占得不多,想來想去可能還是大家的念力有幫忙到吧! :) 謝謝大家!! 另外,從結果來看,我來做個事後諸葛好了,其實 24 小時之內,我們真的能做出比別人做三個月、三年的 library 強嗎? 我不大相信... 這已經不叫「重新發明輪子」,而叫「一小時學會做方的輪子」.. 哈哈 所以我覺得就這次時間有限制的狀況而言,整合別人的智慧結晶,是對的.. 雖然做得很心虛,得了獎也沒有太大成就感, 但是至少可以覺得自己還會做整合工作,能整得乾淨、漂亮... 也算不錯。 在此必須、一定、要向那幾組有重新寫輪子的人致敬!! 我相信給您們三天、三個月、三年,你們也可以做出更讚的 lib! 您們有 guts! 並沒有輸! 您們很讚! ==完==
HBS	這次的 比賽 成了 Porting Game. 不過在這裡 壓縮程式 有上層的作法 據我所知 有人用 Typ Checking 什麼檔用什麼 compress algorithm. 但我 用的方法為 假設 你有 K 個 Compress library (k = 1..3) 你要找 最好的 Compress 你可以 Pre-Compress-Test 也就是 先用 K compress algorithm Test 100K (small part of source data) 看那個 algorithm compress rate 最好 就用 那一個 algorithm 去 compress source data. 也就是說 如果 XXX.wav rename 成 xxx.yyy 時 Type Choice compress algorithm 就沒則了
Sorry - reply HBS Dear hbs,	使用 pre-compress-test 是個不錯的方法,不過基本上只要對 Phantom 稍有認識就知道,他絕對不可能讓 Word 檔就叫 .doc,執行檔叫 .exe . . . 。所以 type checking 是一定不可以用副檔名來做的。 我們這組用的是比較陽春的方法,用檔頭來決定,所以即使把 WAV 檔改名成 .hbs,一樣會用 wav 專用的 library 去壓。(所幸 Phantom 沒有連檔頭都動手腳,感謝他)。
Phantom	騙我錢的人不會有好下場的