

# 數字字謎的解謎與字謎產生器的研究

李立中 · 林順喜

摘要：數字字謎遊戲向來是許多人喜歡作的益智性思考遊戲，可以讓人應用推理性的思考，在解題的過程中得到樂趣，並得到解開謎題的成就感。在過去這些字謎都是由專家構思建構出來的，在本文中我們考慮結合英文單字及數字的字謎。在英文單字中代表數字的單字有許多，如 one、two、three、twenty、thirty、hundred、thousand、million 甚至到 billion 等 32 個單字，在本文中，我們將先試著解一個字謎，並嚐試著利用電腦把這些單字作排列組合，使其不只在英文意義上符合其數字之值，而各字母以不同的數字代入也能產生合理的運算式，並且利用電腦來驗證產生的字謎是否有解。除此之外，這個程式也可以用來解其它的字謎。如此一來，我們也可以由程式產生各式各樣不同的字謎，可以供人作思考解謎之用。程式執行出來的結果，已經找到了四層以下的所有有解的字謎，至於五層以上，由於複雜度太高且實用性較低，人腦要解已難以負荷，所以便不再繼續窮舉，但是我們的程式仍可產生及求解。

## 一、緒論

### 甲. 字謎的介紹：

字謎的花樣是千奇百怪，例如一個英文句子「Give me money.」我們就可以把它化成為「GIVE × ME = MONEY」這道式子。這樣的謎語有趣的地方在於，我們不是隨便解一道式子，如此和平常解  $x, y, z$  的代數式子豈不是沒有差別，這樣的式子，除了在字面上有它的意義，而用數字代入之後，又有它另一層的意義（數學上的意義）。這是字謎有意思的地方，人會想要找到它另一層的意義。

再者，這樣的字謎，解答通常不只一組，又因為光是一道式子所給的資訊並不完全的，所以解法更是依各人不同，當然，解出來的答案也會不一樣，這樣可以考驗及磨練自己的思考推理能力，看看自己是不是能夠從有限的線索裡找出答案。

相信大家或許有見過這樣的字謎，以每個英文單字為一行，排成直式的加法，如下面這運算式：

$$\begin{array}{r} \text{FIVE} \\ +\text{FOUR} \\ \hline \text{NINE} \end{array}$$

而若是把式子中的各個英文字母以不同的數字代入的話，那麼這個式子也會變成一個有意義的算式，如下面的式子：

以  $F = 1, I = 2, V = 5, E = 4, O = 9, U = 8, R = 0, N = 3$  代入

$$\begin{array}{r} 1254 \\ +1980 \\ \hline 3234 \end{array}$$

當然這只是其中的一組解，其它還會有許多組可能的數字代入會使得這式子還是既符合英文字面上的意義 ( $\text{FOUR} + \text{FIVE} = \text{NINE} \Rightarrow 4 + 5 = 9$ )，也符合代入數字後的運算式 ( $1254 + 1980 = 3234$ )。另外，雖然在上例中若是把  $F=0$  代入其中，也許可以找到另一組解，但是這種第一位數為0的，我們也把它排除在外，以免猜字謎時產生困擾。而本文中所介紹的首先便是找出所有在英文意義上能符合的組合，接下來便是利用電腦來驗證看看是否這些組合都至少有一組數字能夠代入後仍然得到一個正確的式子。

當然若是  $\text{FIVE} + \text{FOUR} = \text{THREE}$  也可能可以找到一組數字代入使得數字算式是正確的，然而，在英文上的意義  $5 + 4 = 3$  顯然是不合理的，這種的組合，我們目前不列入考慮。在過去這些字謎大都由專家（稱為出題者）以人腦構思建立出來。在本文中我們設法讓電腦來達成這樣的任務。

而在驗證字謎是否有解的演算方法，便是電腦解字謎的方法。因此，利用這種演算方

法，即使是任何字句中的單字組合成的一組式子，也能馬上求出這種組合是不是也有一組合理的數字解。

## 乙. 人解字謎和電腦解字謎的差別：

人在解這種字謎的時候，相信有學過基本代數的人首先都會想到試著利用代數來列出各個字母間的關係，當然，如果只就單純的各位數字間的算數關係。如前例  $\text{FIVE} + \text{FOUR} = \text{NINE}$  中，我們可以找出  $2F = N$  (在沒有進位的情況) 或  $2F +$  右邊的進位  $= N$  (進位的情況)，其它幾位的數字大概也可以用這些情況來討論，列出了這些關係以後，再代入一些數字到各個字母裡去，這樣子在代入數字的時候便能較快的判斷自己猜測的數字是不是正確的。

或者更多人反正是爲了好玩而猜字，就會嚐試著直接將數字代入到式子中，這樣子就是用嚐試錯誤的方法，其實道理和前面的方法一樣，只不過他是直接代入來測試合不合理，而不是利用關係式求解。然而這種方法，可能會多帶入一些不適合的解而不自知，較沒有效率。

而電腦在解這些字謎的時候，大致上就是以嚐試錯誤的方法，從0開始代入，全都帶入之後，便計算看看合不合理，如此重複下去，便能找到解，當然，同樣的事情，電腦作起來只要短短的幾秒就可以完成了，不過，人會在解謎中得到樂趣和成就感，這是電腦所沒有的吧！

在下文中我們將會試著用一些方法來解一個字謎。讀者可以參考看看，也許會有更好

更快的解法來解謎。也可以比較一下，人腦和電腦解謎方法上的不同。

## 二. 動動腦筋來解題

我們來試試看「SEND + MORE = MONEY」。首先我們不打算以列出所有關係式的方法，因為關係式並不好列，再加上考慮進位的問題，那列出來的方程組太多了，反而容易混淆我們。另一方面，這種字謎可能會有多組解答，所以我們考慮到這些方程組並沒有提供完整的資訊，而且因為有多組解答，因此我們如果能夠先假設其中一些字母的數字，也許就能推出一組合理的答案。

我們打算以代入數字的方法，但是要從哪一個字開始代呢？我們先來統計一下這道式子的字母吧。

字母	S	E	N	D	M	O	R	Y
出現次數	1	3	2	1	2	2	1	1

從這個表格看來，「E」出現的次數最多，所以我們要是先由「E」開始代入我們假設的數字的話，那麼，剩下的未知字母就會最少。但是，我們再看看它們的排列位置：

$$\begin{array}{r} S \boxed{E} N D \\ + M O R \boxed{E} \\ \hline M O N \boxed{E} Y \end{array}$$

我們發現這三個「E」在位置上分別在不同行，如此一來，我們並沒有辦法因為代入「E」之後，而能利用這些假設的數字得到其

它的字母可能會是什麼數字。接下來我們有二種選擇，一是繼續代入其它數字到其它出現次數較多的字母，另一就是我們改用別的方法。

我們試著代入「N」、「M」、「O」等出現過二次的字母，當我們試著第二個要以「M」代入數字的時候我們想到了可以利用另一條規則。因此我們決定改走另一條路。

因為式子的每一行的第一個字母不應該為0，所以我們知道「S」和「M」都應該是大於0的數字。接下來我們知道二個一位數字相加最大是18，又因為每個字母要求是不一樣的數字，因此我們知道「S + M」最多等於17，這時候「M」的答案已經出現了，「M」就只有可能是1。既然「M = 1」，又由式子裡「S + M」(也就是 S + 1) 必須有進位，才能符合「MONEY」的「M」也會等於1，所以「S」就一定是8或是9 (但是其實應該是9，原因在下面一段)，然而同時我們就可以知道，「O」必定會是0。如下：

$$\begin{array}{r} \boxed{9} E N D \\ + \boxed{10} R E \\ \hline \boxed{10} N E Y \end{array}$$

$$M = 1, S = 9, O = 0.$$

解出了這幾個字母的數字之後，我們再回到之前出現次數最多的「E」代入其它沒用過的數字，此時，未知的字母減少了，剩下的數字也減少了，而我們的線索則更多了。

看到上面的式子，我們發現「E + 0 = N」，還記得二個字母的數字不會相同嗎？還

記得二個一位數字加起來最多進位只會是1嗎？所以我們知道「 $N = E + 0 + 1$  (進位的1)」。(這也就是為什麼「 $S$ 」一定是9, 因為「 $E + O = N$ 」不會再有進位, 所以「 $S$ 」一定要是9) 有了這條線索, 我們還知道, 「 $N + R = 1E$ 」或「 $N + R + 1 = 1E$ 」。

再看剩下2到8的數字中, 嚐試用連續的二個數字放入「 $E$ 」和「 $N$ 」中, 答案已經呼之欲出了。例如以「 $E = 2$ 」和「 $N = 3$ 」代入, 則我們會發現「 $R = 9$ 」, 顯然不對, 因為9已經用過了。這時我們又找到一條線索, 就是既然「 $E + 0 + 1 = N$ 」, 又「 $R$ 」不是9, 所以我們只能接受「 $N + R + 1 = 1E$ 」這條路。如此一來, 「 $R = 8$ 」就肯定了。

$$\begin{array}{r} \boxed{9} \text{ E N D} \\ + \boxed{108} \text{ E} \\ \hline \boxed{10} \text{ N E Y} \end{array}$$

$M = 1, S = 9, O = 0, R = 8$ 。

剩下的字母間並沒有什麼特別的關聯性了, 所以我們可以把「 $E$ 」和「 $N$ 」以連續的二個數字 (現在只能在2到7中尋找了) 代入, 剩下的數字中再找個給「 $D$ 」, 最後再看看「 $Y$ 」所得的數字是不是沒有出現過的數字就行了。當然, 我們要記得「 $D + E = 1Y$ 」(因為必須要有進位才能符合「 $N + 8 + 1 = E$ 」)

既然如此, 我們試著讓「 $D$ 」則由7開始, 如此一來, 顯然「 $E$ 」和「 $N$ 」必須是5和6, 所得到的「 $Y$ 」會是2, 才符合了我們進位和不重覆數字的要求。如下:

$$\begin{array}{r} 9567 \\ +1085 \\ \hline 10652 \end{array}$$

$S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2$ 。

這是我們嚐試著以「SEND + MORE = MONEY」為例來解謎, 我們可以發現, 有些時候一開始的直覺不一定正確 (例如一開始我們嚐試用關係式來解代數、或是一開始想由出現最多次的「 $E$ 」開始解起之類的直覺), 不妨由其它方向試著來解決問題。

「GIVE + MORE = MONEY」是另一個類似的字謎也是很有趣, 解法雖然和上面的例子相似, 但是其中也會因為字母不一樣而有不一樣的地方, 讀者不妨試試。

### 三. 電腦解題之演算法

字謎的種類當然是各式各樣的、無所不有。例如上文中的例子 SEND + MORE = MONEY, 是以一個有意義的英文句子組成的。有字面上的意義, 並且能夠把字母代入數字, 讓它有另一層意義, 這是字謎有趣也是它吸引人的地方。但是有意義的句子非常的多, 所以我們先從其中的一個子集開始作起, 這也是下文所將討論的, 以英文數字為範圍, 來產生有意義的字謎。

#### 甲. 產生數字字謎的演算法:

1. 列出所有英文數字單字, 不包含複合單字, 共32個, 列表如下: **ONE、TWO、THREE、FOUR、FIVE、SIX、SE-**

VEN、EIGHT、NINE、TEN、ELEVEN、TWELVE、THIRTEEN、FOURTEEN、FIFTEEN、SIXTEEN、SEVENTEEN、EIGHTEEN、NINETEEN、TWENTY、THIRTY、FORTY、FIFTY、SIXTY、SEVENTY、EIGHTY、NINETY、HUNDRED、THOUSAND、MILLION、BILLION 共 32個單字。

2. 決定要產生  $N$  層的字謎。其中  $N$  共用到  $N$  個單字，形成加法或乘法之數謎。例如：

$$\begin{array}{r} \text{FIVE} \\ +\text{FOUR} \\ \hline \text{NINE} \end{array}$$

此例即為  $N = 3$

3. 前  $(N - 1)$  層分別選出  $(N - 1)$  個單字作出各種組合。在排列時，第  $k$  個英文字代表數大於等於第  $(k + 1)$  個數。所以有了  $\text{FIVE} + \text{FOUR} = \text{NINE}$  的組合，便不會再重複選出  $\text{FOUR} + \text{FIVE} = \text{NINE}$  的組合，因為這兩組其實是一樣的，沒有必要重複的驗證。
4. 依照 2. 中選出的單字的對應的數值作運算，看看加起來或乘起來會等於多少，若是這數值正好對應到某一個單字，那麼便找到一組可能的組合。(例如： $\text{FIVE} + \text{FOUR} = \text{NINE}$ ) 若是找不到對應的單字，那麼這組便不是我們要的。(例如： $\text{THREE} \times \text{SEVEN} = \text{TWENTY-}$

$\text{ONE}$ ,  $\text{TWENTY-ONE}$  為複合單字，不在 32 個單字中，所以放棄這種組合。)

5. 找到 3 中的組合後，便開始判斷在這種組合裡，是不是小於等於十個英文字母 (如  $\text{FIVE} + \text{FOUR} = \text{NINE}$  中有 F、I、V、E、O、U、R、N 等 8 個)，因為大於十個字母，就會有字母必須代入重複的數字，這便不合我們字謎的要求。
6. 在排除了 4 中不合理的組合之後，仍可以經由單字的長度來判斷這些組合是否有可能。例如：4 位數加 4 位數得到的和也必定是 4 位數或者是 5 位數。所以若是和的單字長度太長太短都不對。而像 3 位數乘以 3 位數，所得的積必定是 5 位或 6 位數字。(因為其乘積必大於  $100 \times 100 = 10000$ ，而小於  $1000 \times 1000 = 1000000$ ) 相同的，3 位數乘以 4 位數再乘以 3 位數的積則可能是 8 位、9 位、10 位。(原因類似前面所述)。如此便又可以除去許多不可能的組合。
7. 最後以下一節所述之猜字謎的演算法，來判斷剩下的組合，這樣便可以最快的時間找到有解的組合。

## 乙. 猜字謎的演算法：

以  $\text{FIVE} + \text{FOUR} = \text{NINE}$  為例。

1. 找出這組字謎有幾個字母。(找到 F、I、V、E、O、U、R、N 等 8 個)
2. 將數字代入字母。(從 0, 1, 2, ..., 9, 分別代入各個字母)
3. 判斷這個數字是否和其它字母相同，又若是這個字母是第一位，則不能為 0。

4. 重複2,3的作法, 直至所有加數和被加數的字母 (或者所有乘數、被乘數的字母), 已經都有代入數字。(如這裡代到 *F, I, V, E, O, U, R* 都有值即可。)
5. 將剛才代入的數值從最低位到最高位作運算, 運算出來的結果, 再看看最後一行的單字, 若是這一位的字母已經在上面幾行出現過 (如*I, E*在 *FOUR, FIVE* 中出現過), 則可以直接比對這個算出來的值是否相同, 相同就繼續作下去, 不同就表示這組數字不合。若是這一位字母未在上面出現過 (如 *N*), 再把計算的結果比對看看這數字是否有被用過, 如果沒有就

直接代入, 作為這字母的值。若是有的話, 則表示這組數字不合。

6. 找到一組合的就表示這組字謎有解, 否則就是無解的字謎。

#### 四、程式執行結果與分析

本程式使用 C++ 語言寫的, 在Tur- bo C++或Borland C++ Builder等編譯器下編譯皆可, 將這程式在 Pentium 100、32MB RAM 的電腦上試著跑了一遍, 經過產生字謎程式部分所產生有可能的字謎, 產生的結果如下表: (其中最後一層為和或積)

層數	加法			乘法		
	可能有解組數	有解組數	執行時間	可能有解組數	有解組數	執行時間
3	69	6	5分 50秒	10	1	4分 30秒
4	137	36	11分 25秒	3	0	1分 15秒
5	209	61	17分 15秒	0	0	小於 1秒

其中可能有解的組合的意思是, 經過初步簡單的條件, 即在前面提到的 (甲) 產生字謎的演算法中的判斷後, 所產生的組合, 但是尚未經過實際求解, 所以可能會有如 *FIVE + THREE = EIGHT* 在產生字謎的演算法中無法判斷其是否無解的組合。經過實際的求解過程, 除去這種無解的可能, 則是我們真正要的字謎, 也就是上表中「有解組數」之值。

至於解出來有解的這些字謎, 附在本段的結尾, 供大家參考, 從表中看出加法要符合條件比較容易, 而乘法則是到5,6層連有可能

的組合都找不到了, 因此只發現了一組乘法數謎有解! 話說回來, 要猜出乘法的字謎可比加法麻煩多了。而大家也許會發現, 不是本來說有加減乘除的, 但是怎麼這裡只有作加法和乘法, 這是因為加法的順序改變, 便可以成為連減的組合 (把和當成被減數, 其它的則是減數, 剩下一個就是差了), 而乘法的順序改一改, 也是可以成為除法 (積作為被除數, 乘數或被乘數都可以當除數和商)。

以下就是我們所列出來所有有解的字謎組, 讀者可以向自己的思考推理能力挑戰。

乘法: (只有一組)

(1)	SIX	965
	× TWO	× 218
	TWELVE	210370

加法:

三層的組合: 共6組

(1)	FIVE	1254	(2)	FOUR	5238	(3)	FOUR	2503	(4)	FOUR	1230	(5)	TWO	734
	+ FOUR	+ 1980		+ FOUR	+ 5238		+ THREE	+ 69311		+ ONE	+ 256		+ TWO	+ 734
	NINE	3234		EIGHT	10476		SEVEN	71814		FIVE	1486		FOUR	1468
(6)	ONE	206												
	+ ONE	+ 206												
	TWO	412												

四層的組合: 共36組

(1)	FIFTY	76749	(2)	FORTY	83457	(3)	FORTY	96342	(4)	FORTY	37984	(5)	FORTY	29786
	TWENTY	485149		FORTY	83457		THIRTY	410342		FIFTEEN	3238550		TEN	850
	+ TEN	+ 451		+ ZERO	+ 2143		+ TEN	+ 458		+ FIFTEEN	+ 3238550		+ TEN	+ 850
	EIGHTY	562349		EIGHTY	169057		EIGHTY	507142		SEVENTY	6515084		SIXTY	31486
(6)	FORTY	21346	(7)	FORTY	31206	(8)	FORTY	12834	(9)	THIRTY	593250	(10)	TWENTY	185710
	NINE	5750		SIX	958		FIVE	1650		TWENTY	546750		TEN	157
	+ ONE	+ 150		+ FOUR	+ 3142		+ FIVE	+ 1650		+ TWENTY	+ 546750		+ ZERO	+ 6543
	FIFTY	27246		FIFTY	35306		FIFTY	16134		SEVENTY	1686750		THIRTY	192410
(11)	TWENTY	120416	(12)	TWELVE	126406	(13)	ELEVEN	858682	(14)	NINE	3538	(15)	SEVEN	68782
	FIVE	8750		NINE	5756		SEVEN	78682		EIGHT	85291		SEVEN	68782
	+ FIVE	+ 8750		+ NINE	+ 5756		+ TWO	+ 930		+ THREE	+ 19488		+ SIX	+ 650
	THIRTY	137916		THIRTY	137918		TWENTY	938294		TWENTY	108317		TWENTY	138214
(16)	SEVEN	91316	(17)	SEVEN	82524	(18)	SIX	206	(19)	FIVE	3025	(20)	FIVE	5091
	FOUR	4872		THREE	19722		TWO	394		FOUR	3496		THREE	78411
	+ ZERO	+ 5128		+ TWO	+ 106		+ ONE	+ 415		+ ZERO	+ 1564		+ THREE	+ 78411
	ELEVEN	101316		TWELVE	102352		NINE	1015		NINE	8085		ELEVEN	161913
(21)	FIVE	1329	(22)	FIVE	1036	(23)	FIVE	9021	(24)	FOUR	1650	(25)	FOUR	2657
	THREE	86499		TWO	495		TWO	846		FOUR	1650		FOUR	2657
	+ ZERO	+ 5940		+ TWO	+ 495		+ ONE	+ 671		+ ONE	+ 632		+ ZERO	+ 8176
	EIGHT	93768		NINE	2026		EIGHT	10538		NINE	3932		EIGHT	13490
(26)	FOUR	1765	(27)	FOUR	8637	(28)	FOUR	8925	(29)	THREE	32066	(30)	THREE	23577
	THREE	39544		TWO	946		TWO	739		THREE	32066		THREE	23577
	+ ONE	+ 784		+ TWO	+ 946		+ ONE	+ 940		+ TWO	+ 391		+ ONE	+ 817
	EIGHT	42093		EIGHT	10529		SEVEN	10604		EIGHT	64523		SEVEN	47971
(31)	THREE	29600	(32)	TWO	108	(33)	TWO	235	(34)	TWO	172	(35)	TWO	207
	TWO	254		TWO	108		TWO	235		TWO	172		ONE	764
	+ TWO	+ 254		+ TWO	+ 108		+ ONE	+ 576		+ ZERO	+ 3862		+ ONE	+ 764
	SEVEN	30108		SIX	324		FIVE	1046		FOUR	4206		FOUR	1735

(36)	
TWO	145
ONE	586
+ ZERO	+ 9635
THREE	10366

五層的組合: 共61組

(1) SIXTY 31650 TWELVE 582742 ELEVEN 272429 + SEVEN + 32429 NINETY 919250	(2) SIXTY 45783 NINE 6561 EIGHT 15028 + THREE + 82911 EIGHTY 150283	(3) SIXTY 0763 EIGHT 10286 SIX 907 + SIX + 907 EIGHTY 102863	(4) SIXTY 56237 SEVEN 51819 SEVEN 51819 + SIX + 562 EIGHTY 160437	(5) FIFTY 64682 FORTY 69582 ZERO 3759 + ZERO + 3759 NINETY 141782
(6) FIFTY 34351 THIRTY 564051 SEVEN 92827 + THREE + 56022 NINETY 747251	(7) FIFTY 93958 THIRTY 523458 FIVE 9371 + FIVE + 9371 NINETY 636158	(8) FIFTY 86810 THIRTY 176410 ZERO 3245 + ZERO + 3245 EIGHTY 269710	(9) FIFTY 76734 TWENTY 301534 ELEVEN 181215 + NINE + 5651 NINETY 565134	(10) FIFTY 75732 TWENTY 364832 NINE 8584 + ONE + 984 EIGHTY 450132
(11) FIFTY 87840 TWENTY 425940 EIGHT 57134 + TWO + 426 EIGHTY 571340	(12) FIFTY 30316 TWENTY 172516 FIVE 3042 + FIVE + 3042 EIGHTY 208916	(13) FIFTY 40439 FOUR 4687 FOUR 4687 + TWO + 326 SIXTY 50139	(14) FORTY 46823 FORTY 46823 TEN 291 + ZERO + 7986 NINETY 101923	(15) FORTY 73092 FORTY 73092 FIVE 7654 + FIVE + 7654 NINETY 161492
(16) FORTY 42756 FORTY 42756 ZERO 9172 + ZERO + 9172 EIGHTY 103856	(17) FORTY 56031 THIRTY 378031 EIGHT 48273 + TWO + 396 EIGHTY 482731	(18) FORTY 34782 THIRTY 860782 FIVE 3059 + FIVE + 3059 EIGHTY 901682	(19) FORTY 15946 TWENTY 430846 TWENTY 430846 + TEN + 408 NINETY 878046	(20) FORTY 23570 TEN 794 ZERO 1953 + ZERO + 1953 FIFTY 28270
(21) FORTY 12390 SIX 856 TWO 972 + TWO + 972 FIFTY 15190	(22) FORTY 10542 FOUR 1065 FOUR 1065 + TWO + 470 FIFTY 13142	(23) THIRTY 418943 THIRTY 418943 TWENTY 407543 + TWENTY + 407543 HUNDRED 1652972	(24) THIRTY 159018 THIRTY 159018 TWENTY 176418 + TEN + 164 NINETY 494618	(25) THIRTY 128915 THIRTY 128915 TWELVE 174364 + EIGHT + 48021 EIGHTY 480215
(26) TWENTY 109416 TEN 194 ZERO 8953 + ZERO + 8953 THIRTY 127516	(27) TWENTY 126913 NINE 9596 ONE 896 + ZERO + 7608 THIRTY 145013	(28) ELEVEN 303538 ELEVEN 303538 FIVE 4953 + THREE + 67133 THIRTY 679162	(29) ELEVEN 696563 NINE 3436 NINE 3436 + ONE + 836 THIRTY 704271	(30) ELEVEN 797275 NINE 5057 FIVE 4027 + FIVE + 4027 THIRTY 810386
(31) ELEVEN 151814 EIGHT 17602 EIGHT 17602 + THREE + 20311 THIRTY 207329	(32) ELEVEN 494046 THREE 73544 THREE 73544 + THREE + 73544 TWENTY 714678	(33) NINE 3239 SEVEN 49793 SEVEN 49793 + SEVEN + 49793 THIRTY 152618	(34) EIGHT 64391 EIGHT 64391 EIGHT 64391 + SIX + 842 THIRTY 194015	(35) EIGHT 68071 EIGHT 68071 TWO 135 + TWO + 135 TWENTY 136412

(36)	(37)	(38)	(39)	(40)
SEVEN 60902	SEVEN 35752	SIX 124	SIX 358	SIX 109
SEVEN 60902	SEVEN 35752	TWO 308	ONE 721	ONE 285
FIVE 7590	THREE 16855	ONE 879	ONE 721	ZERO 7532
+ ONE + 820	+ THREE + 16855	+ ZERO + 5968	+ ONE + 721	+ ZERO + 7532
TWENTY 130214	TWENTY 105214	NINE 7279	NINE 2521	SEVEN 15458
(41)	(42)	(43)	(44)	(45)
FIVE 1409	FIVE 1235	FIVE 2601	FIVE 8061	FIVE 2743
THREE 87399	TWO 740	TWO 358	ONE 731	ONE 653
ZERO 2935	TWO 740	ONE 841	ONE 731	ONE 653
+ ZERO + 2935	+ ZERO + 6580	+ ONE + 841	+ ONE + 731	+ ZERO + 9386
EIGHT 94678	NINE 9295	NINE 4641	EIGHT 10254	SEVEN 13435
(46)	(47)	(48)	(49)	(50)
FOUR 1408	FOUR 2839	FOUR 8769	FOUR 1256	THREE 63511
FOUR 1408	FOUR 2839	THREE 45911	TWO 972	THREE 63511
ONE 453	ZERO 5198	THREE 45911	TWO 972	THREE 63511
+ ZERO + 2384	+ ZERO + 5198	+ ONE + 721	+ ONE + 230	+ TWO + 684
NINE 5653	EIGHT 16074	ELEVEN 101312	NINE 3430	ELEVEN 191217
(51)	(52)	(53)	(54)	(55)
THREE 20944	THREE 12644	THREE 29733	THREE 16955	THREE 29611
THREE 20944	THREE 12644	TWO 248	TWO 103	TWO 245
TWO 217	ONE 794	TWO 248	TWO 103	ONE 581
+ ZERO + 3497	+ ZERO + 8467	+ ONE + 863	+ ZERO + 8593	+ ONE + 581
EIGHT 45602	SEVEN 34549	EIGHT 31092	SEVEN 25754	SEVEN 31018
(56)	(57)	(58)	(59)	(60)
TWO 102	TWO 109	TWO 138	TWO 102	ONE 326
TWO 102	TWO 109	ONE 856	ONE 295	ONE 326
ONE 235	ZERO 2369	ONE 856	ONE 295	ONE 326
+ ONE + 235	+ ZERO + 2369	+ ONE + 856	+ ZERO + 6542	+ ONE + 326
SIX 674	FOUR 4956	FIVE 2706	FOUR 7234	FOUR 1304
(61)				
ONE 423				
ONE 423				
ONE 423				
+ ZERO + 9364				
THREE 10633				

## 五. 結論

### 1. 目前的結論:

目前在五層以下的單純的加乘運算的所有組合已經全部計算過了，總共就只有附表列出來的那些單字的組合，能有合理的數字解法，而單純的減算和除算的單字組合則可以由加乘運算得到（這之間是1對1而且映成的對應，所以作完加乘運算的組合，等於作完減和除的運算），所以說這部分的組合已經全部找出來了。

### 2. 未來發展方向:

將來的發展方向不在於往五層以上的字謎發展，因為五層以上的字謎已經不大可能由人來解，所以將來可以發展低層數的，大約二、三層的，而改由有意義的英文字句組合（例如： $HE \times WAS = HERE$ ），甚至可以由中文的詩句組合（例如：年年 $\times$ 歲歲=花相似，歲歲/年年=人/不同，二個式子聯立），或者是四則混合運算的字謎來發展，如此一來，相信猜這些字謎會更加的有趣的。而應用這種方式，我們可以找出很多種字謎來作為大

家娛樂及磨練自己思考推理能力之用。也可以自己出謎, 利用我們的程式, 來驗證看看自己出的字謎是不是有解, 可以拿去給同學、朋友動動腦。

## 六. 參考文獻

1. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics: an applied introduction, 3rd ed, Addison Wesley Publisng Company, 1994.
2. Wackerly & Mendenhall & Scheaffer, Mathematical Statistics with Applications, 5th Edition, Wadsworth Publishing Company, 1996.
3. Richard E. Neapolitan & Kumarss Naimipour, Foundations of Algorithms, D. C. Heath and Company, 1996.
4. Ellis, Horowitz & Sartaj Sahni, Fundamentals of Data Structures in Pascal, 4th Edition, Computer Science Press, 1994.
5. Otto Bretscher, Linear Algebra with Applications, Prentice Hall, 1997.
6. Stephen Prata, C++ Primer Plus, 2nd Edition, The Waite Group, Inc, 1995.
7. 洗鏡光, 名題精選百則技巧篇 — 使用 C 語言, 格致圖書公司, 1990.
8. 李維, 精通 Borland C++ Builder — 視覺化 C/C++ 程式設計「基礎篇」, 博碩顧問股份有限公司, 民國86年3月。

—本文作者任教於國立臺灣師範大學資訊教育系—