Homework 5
Input – p5.in, output – p5.out
Deadline – June/8/2018

為降低資料儲存的空間或增加資料傳送的速度，霍夫曼編碼是常用的方法。

假設有一個字元集，每個字元出現的次數是已知的。現在要把每個字元編碼成為一個二元字串(例如把'D'編碼作 110)，採用的編碼必須合乎以下條件：一個字元的編碼不可以是另一個字元的前置(prefix)，因為這樣在解讀編碼時就可以不需要加上「一個編碼的長度」就能解讀出字。前置的定義如下：若一個字串 s1 為另一個字串 s2 的前置，則從 s2 的最後一個字元開始，連續刪除一定數量的字元後可以得到 s1（s2 本身也是 s2 的前置），舉例而言:如果字元'A'的編碼是 110，而字元'B'的編碼為 10，則'B'的編碼不為'A'編碼的前置；如果字元'C'的編碼為 1100，而字元'D'的是 11，則'D'的編碼是'C'編碼的前置。以下的編碼方式可以在符合這個條件下給出最經濟的編碼。

編碼法，**請參考老師投影片「Chap5_final」**

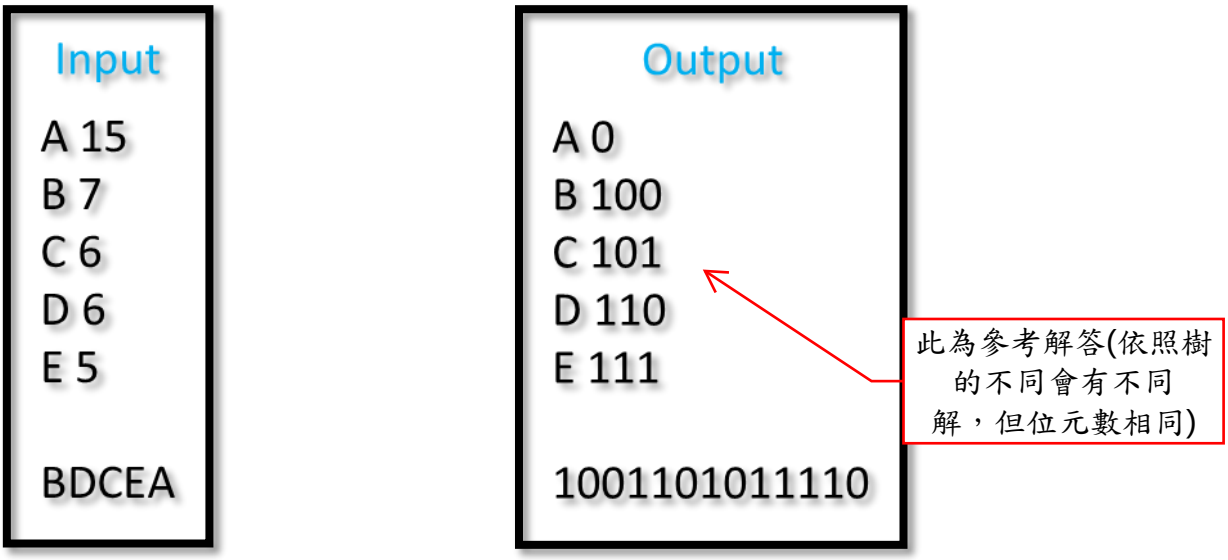
1. 如以下所述建立一棵二元樹：

先從字元集選取兩個出現次數最低的字元作合併，合併後以一個全新的虛擬字元取代這兩個字元，新字元的頻率等於這兩個就字元頻率的總和，並令這兩個就字元為此新字元的兩個子樹，左右不拘。重複以上動作，直至字元集剩下一個字元為止。

2. 並依照以下所述方法將各字元作編碼。

由上一步驟所得之二元樹，將每個內部節點(internal node)連往左子樹的邊(edge)標記為'0'，連往右子樹的邊標記為'1'。一字元的編碼即為從樹根(root)至此字元，經過的每一個邊的標記所成之字串（如：在此'D'編碼作 110）。

本次作業目標在對一串文字(ASCII)進行編碼，用於建樹的各字元出現次數也已提供，輸入、輸出格式如以下：



注意：

1. 將受編碼的文字可能會有「空白」、但不會有「換行」。
2. Input 中字元出現次數之後沒有空白（如：D 6）、介於出現次數與編碼文字中間為一單純「換行」。
3. Output 的字典表與輸出編碼中間亦為單純空行。
4. 有檔案格式問題、請務必詢問助教或是知道的人，這次作業會使用助教的解碼程式解開、若能成功還原才算是對。