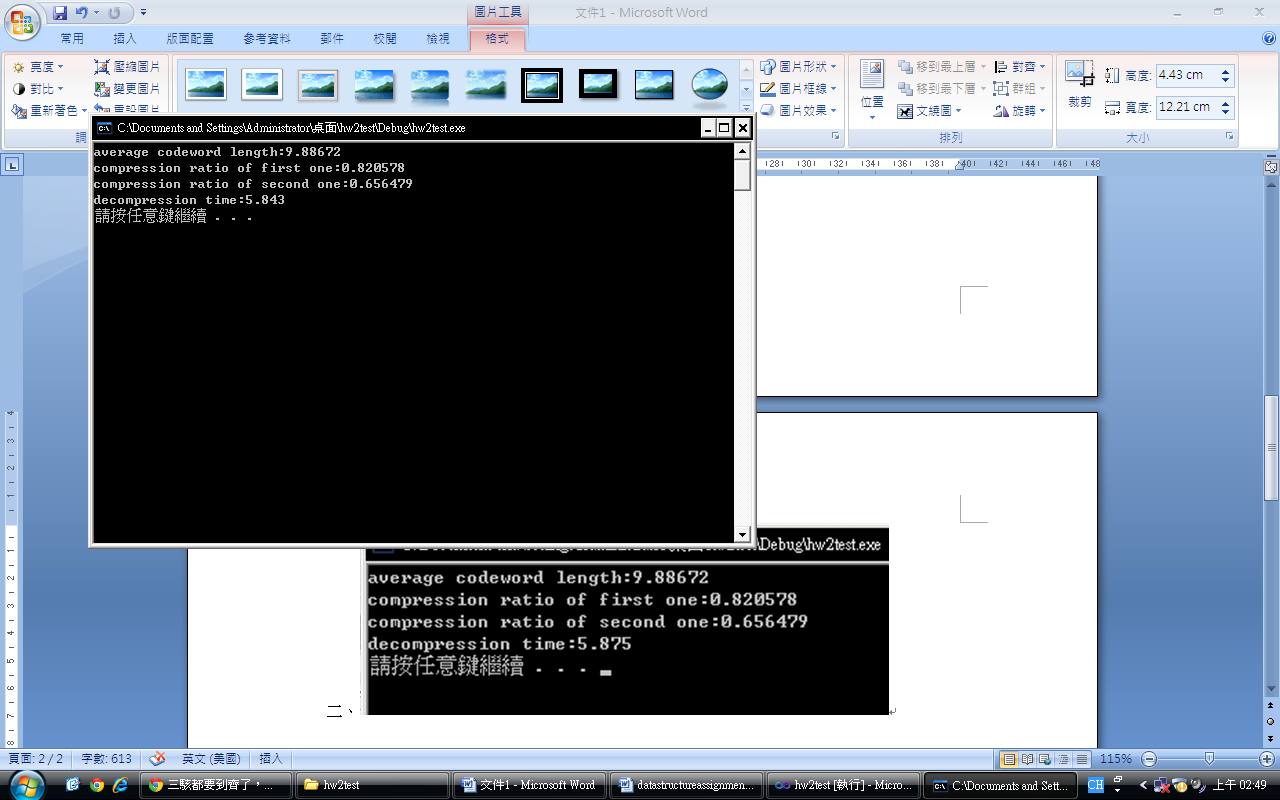
**Data Structure Assignment 2**

**Huffman Compression**

資工A班0016053翁慶年

1. Implementation  
   　　我分別寫了三個class，HufT用來讀取圖片資料並輸出Huffman Table；Encoder利用讀入的Huffman Table將圖片壓縮；Decoder利用讀入的Huffman Table將圖片解壓縮。另外還寫了兩個struct，HTNode是Huffman Tree Node；SymFreq是存放從圖片中讀取的symbol出現次數，symbol為0~255正好直接做為陣列足標對應Count。  
   　　HufT部分，首先HufT讀取圖片資料後，在HufT::Encode中一一放入SymFreq[257]陣列中，SymFreq[0]不使用，並做成minheap，之後每次從陣列中取出兩個最小count的Symbol，使其分別為一個新的HTNode的左右子樹，並且將其在SymFreq中的count設為左右子樹相加，並且丟回陣列中做downheap。這個新的HTNode的symbol沒有什麼特別意義，所以我就從256開始遞增assign給每一個這樣的新節點。  
   　　Huff Tree建立好之後，我寫了一個遞迴Post\_travel，每當走到leaf node時就將走過的路徑記錄下來，使其成為該symbol的codeword。記錄的方法是用static int level計算我走了幾步，static codeword[256]用來存放路徑，最後再存回。  
   　　最後呼叫print將HufT輸出到檔案。  
   　　Encoder部分，首先讀入HufT，將symbol對應codeword。Compression將一次一個byte讀入後，轉成symbol對應的codeword，在輸出到.huf檔之前，我寫了一個buffer來存放，要輸出的codeword每滿八個之後，將這八個視為binary轉成char，在輸出。檔案讀到最後可能不會滿八個，所以我加了一個filler存放codeword length超過8的前7個codeword pattern，用以填充最後buffer沒有滿的情況。  
   　　Decoder部分，HTree首先讀入HufT，將symbol對應codeword後，根據codeword建立Huffman Tree，方法是從root開始依照codeword，0往左子樹、1往右子樹走，是NULL就new HTNode，codeword指示的路徑走完之後，將symbol assign給最後走到的那個leaf node。HTree建立好之後，decomp一次一個byte讀入.huf檔，利用mask做”and”運算解出byte中的八個bits分別為0或1，再走訪Huffman Tree，走到leaf node時即解出symbol。
2. 平均編碼長度：9.88672個bit  
   第一個圖片壓縮率：0.820578  
   第二個圖片壓縮率：0.656479  
   解壓縮兩張圖片的時間：5.843
3. How to run my program? 在Visual C++中開啟按F5就會自動跑出來了。