token.h & tokend.cpp

在 class Operator 內新增U\_SUB與INV，分別對應單元負號運算子與不支援的運算子

新增一個傳入值為一個double、一個Operator class、一個Tag class與一個string的建構子，用來強制設定Token的各個屬性。

新增一個傳入值為一個double的建構子，作為數值運算元的專屬建構子。

新增一個型別為string的屬性op，儲存輸入的運算子字元。

原先傳入值為char的建構子改做為運算元專屬的建構子，根據輸入值設定Operator屬性與 string屬性，若輸入為不支援的運算子Operartor屬性設為INV。

opToString函式根據Operatror的標記以及輸入的字串，決定回傳值。

串流運算子<<根據Tag，選擇輸出Token的運算子或是輸出浮點數。

Tokenize函式首先將輸入字串裡面的空格去除，之後逐字元掃描，若為數字，則一次將所有連續數字字元儲存成字串後轉換成浮點數，以token類別儲存後置入佇列，若為「-」以外的字元，以token類別儲存後置入佇列，若為「-」，先以前後字元分辨此處代表減號、單元負號運算子還是負數，若為第一個字元或前一元為非「)」的運算元，後一個字元為數字，則判斷為負數，處理方式同數字，若後一個字元為其他運算子，則判斷為單元負號，處裡方式同一般運算子，若前一個字元為數字或「)」運算子，則判斷為減號，處理方式同一般運算子。

stack.h

將原先的陣列改為vector應對極長算式，並將其他函式修改以配合vector操作。

新增無傳入值的Top與Delete函式以方便部分操作，免除宣告暫時物件。

queue.h

將原先的陣列改為vector應對極長算式，並將其他函式修改以配合vector操作。

因普通佇列讀寫方式受限，新增dump函式讀出所有資料。

postfix.cpp

傳入值為token的isp函式，依照運算子回傳不同優先值，以講義內的優先順序修改以符合新增的運算子，預設傳回值為INT32\_MAX。

傳入值為token的icp函式，依照運算子回傳不同優先值，以講義內的優先順序修改以符合新增的運算子，預設傳回值為INT32\_MAX。

傳入值為Token的isOperator與isOperand函式，根據屬性回傳1或0，分別代表是與否。

傳入值為char的isOperator與isOperand函式，雖有修改但並未使用，故不介紹。

postfix函式完全參考講義，將部分for條件迴圈的部分改while，一開始先捨去Tokenize輸出佇列最後的運算子，完成轉換後才額外加上。

evaluate.cpp

evaluate函式依序讀出佇列內容，若為運算元則推入堆疊，若為運算子，則根據運算子，取出相對應數量的運算元，執行相對應的運算後將結果存入，若取出元素時發現堆疊為空，表示算式不合理，傳回NAN；若遇到不支援運算元，傳回NAN，最後佇列讀取完成並執行所有運算後，檢差堆疊內是否僅剩一個元素，若否則說明算式不合法，傳回NAN，若僅剩一元素，傳回運算結果。