

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，處理方式同一般運算子，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，檢查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。