

Hw3-2 README

eval.cpp

class deque:

- 繼承自vector是一個類別樣版。
- 會用deque是因為不確定在程式中是否會需要使用到stack還是queue。因此選用deque

bool isNumeric(char * c_string): 判斷字串是否為數字，以便判斷是operator還是operand

double performance(string token, string num1, string num2): 依照token決定 num1和num2該做什麼運算。

deque<string> InputInstructions()讓使用者輸入prefix expression，依照空格來做切割，並將所有字串回傳。

main

- 使用InputInstructions將使用者資料回傳。
- main函式從回傳的字串陣列中判斷，如果連續三個字的排列方式是(不是數字、是數字、是數字)則代表這三個字串可以做運算，這時候就會傳入performance做運算，並將運算結果存回deque中「不是數字」的那個位置，並將後兩個是數字的物件erase。
- 迴圈持續到deque的size == 1
- 最後將deque中唯一一個數字output。

prefix.cpp

class Stack

- 繼承自vector，是一個類別樣板。

bool isNumeric(char * c_string): 判斷字串是否為數字，以便判斷是operator還是operand

int Precedence(string operatr): 回傳operator的precedence。

Stack<string> InputInstructions()讓使用者輸入infix expression，依照空格來做切割，並將所有字串回傳。

Class OperatorStack:

- 繼承自Stack<string>，專門處理Operator的Stack。
- Stack<string>OperatorPush會依照Operator的precedence做pop與push。
 - 在OperatorPush中，新來的operator(in-comming)的precedence如果小於等於top 的precedence，則把top丟進準備回傳的Stack中。迴圈持續到in-comming的precedence > top 的precedence

main

- 先用inputInstruction()讓使用者輸入infix expression。

- ——判斷infix中的字串，如果是數字則push入buffer。如果不是則使用OperatorPush，如果push回傳值的size > 0，代表operatorStack中有precedence較大的operators被pop出來。
- 每次從OperatorStack pop出的operators中pop一個出來，再從buffer中pop出兩個，做字串相加，最後push回buffer。迴圈直到OperatorStack pop 出的operator都用完。
- 當infix expression中的所有字串都scan過，就會將eos push入OperatorStack中。將OperatorStack pop出的Operators依照前一步驟將buffer裡的物件做處理。
- 最後將buffer中所有物件輸出。