Quine-McCluskey method

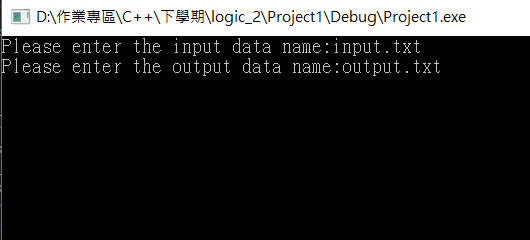
姓名:李明峯

系級:電機110

學號:E24066551

操作手冊

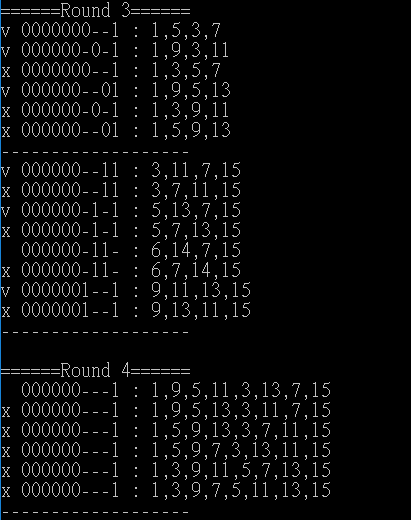
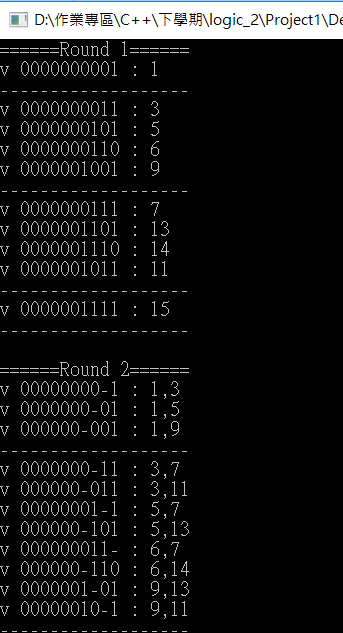
C++:

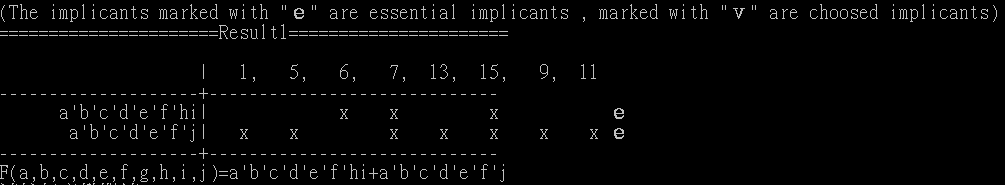
1. 執行程式後鍵入輸入及輸出檔案名稱

input

output

1. 版面上即顯示過程與答案

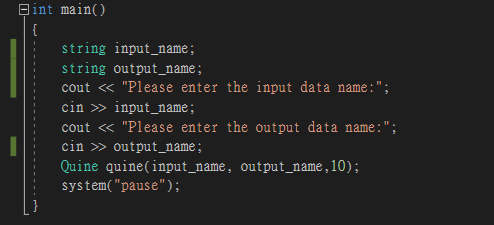




圖上會標記選擇的implicant

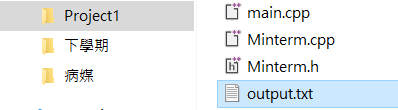
1. 可透過修改Quine物件中的參數設定input的bit數

Ex.10bit(如下圖)

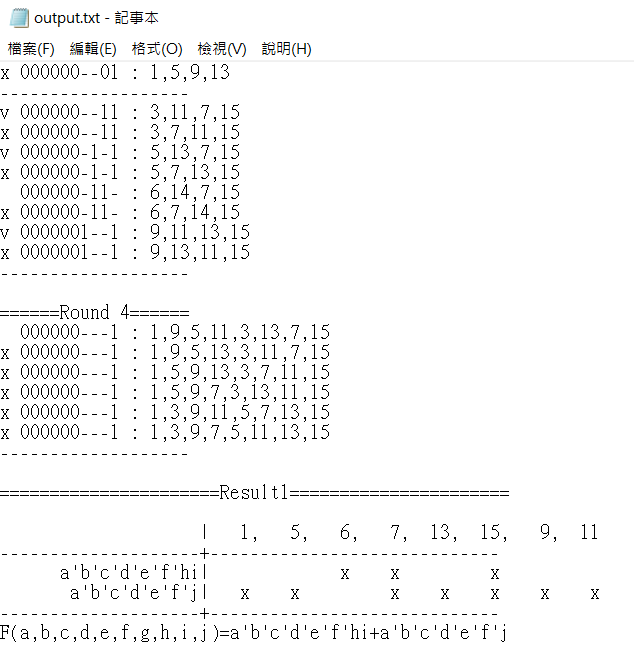


10 bit

1. 在使用者輸入的output位址輸出txt檔



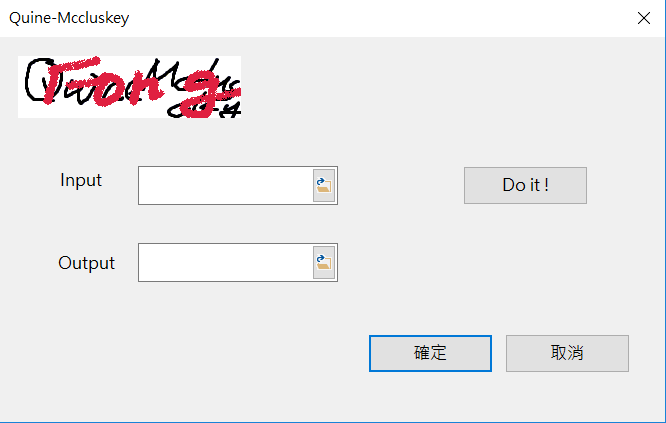
output



內容與程式執行的介面相同

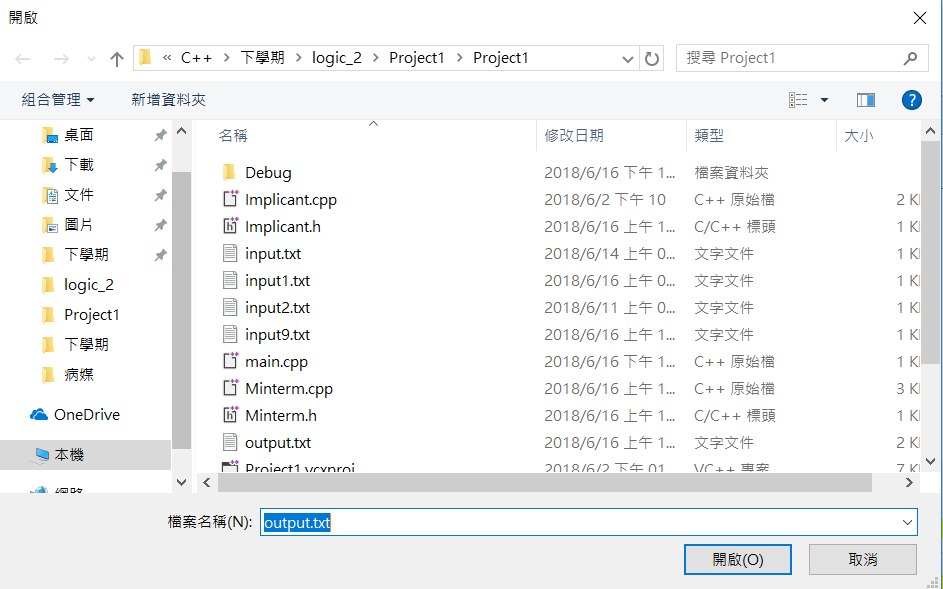
GUI版本:

1. 選取檔案(及設定輸出位置與名稱)

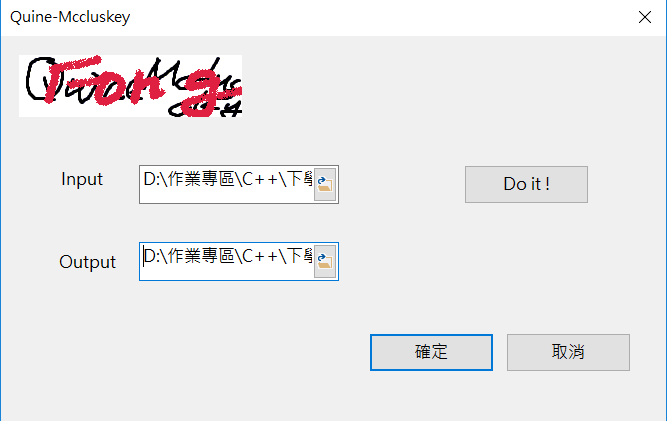


input

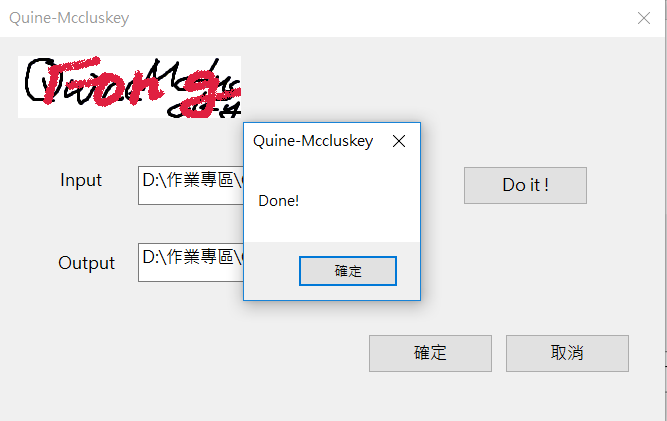
output



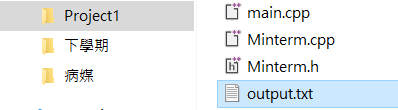
自行輸入名稱+.txt



1. 按下”Do it!”執行程式

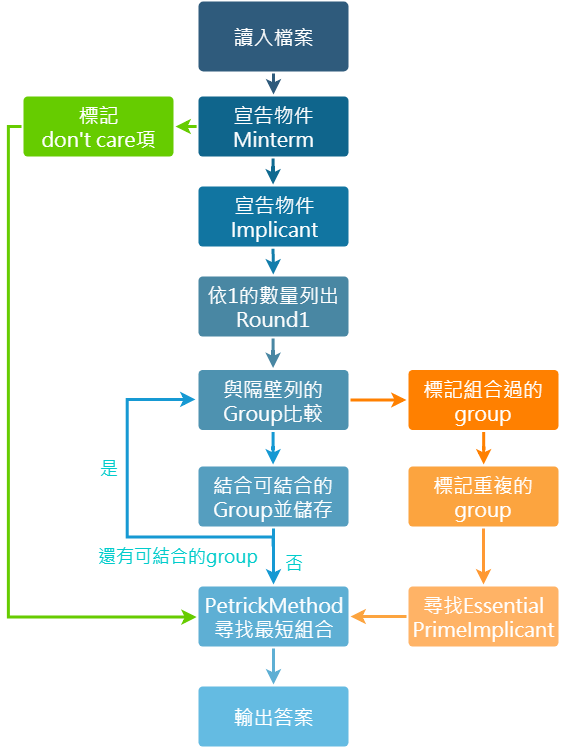


1. 在使用者選擇的output位址輸出txt檔



output

程式撰寫流程



作業心得

第二次的作業是用Quine-mccluskey method將算式化簡，因為我上次的4bit化簡作業也是使用Quine-mccluskey method來寫，我認為這次的程式作業應該是易如反掌，修改幾個細節就好了。但是仔細想想之後，我發覺第一次的作業我因為沒經過詳細的程式架構思考，導致程式的設計及過程十分混亂，很多資料是東拼西湊，造成了程式的維護難度變得得很高。

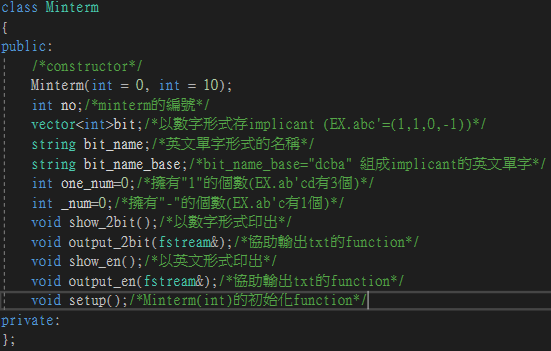
這次作業的難度提升到10bit，給我了一個很好的改進機會，我重新思考了程式的編寫流程，並且重打了一份新的code，有了第一次的經驗之後，第二次的嘗試之後縮短了不少程式碼，也把過去寫死的幾個參數改為可變的參數，大大提升了程式的可運用範圍，給我了不少的成就感。

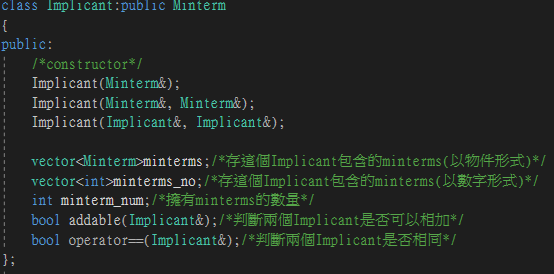
邏設作業給我很棒的機會與理由去嘗試在程式上做各種不同的運用，也讓我稍稍的入門了MFC的編寫方法。相信不管在邏輯設計或者程式設計的學習殿堂裡我都還只是個初入領域的初學者，希望在往後的求學歷程中我可以更加精進，在某個領域裡能夠有所成就。

最後感謝助教這學期對我們提出問題的不吝指教，以及在知識層面上的幫助。

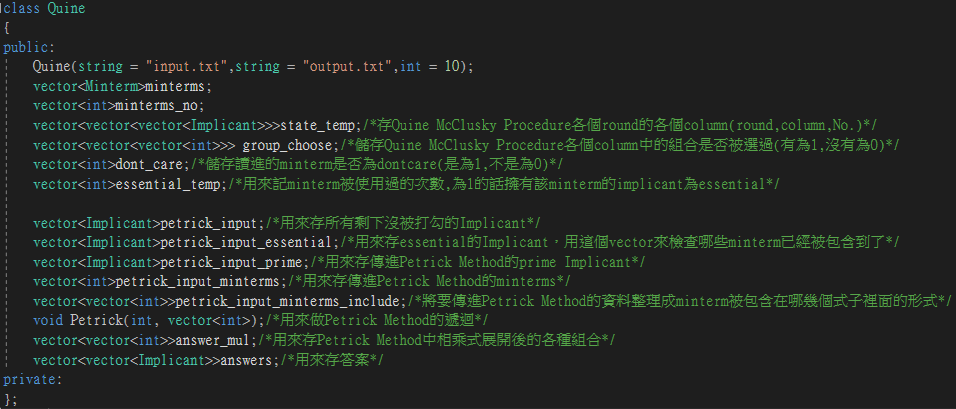
Source code說明

C++:

1. 建立Minterm的class用來儲存不同minterm

建立Implicant的class來存minterms組合而成的group(Implicant) 

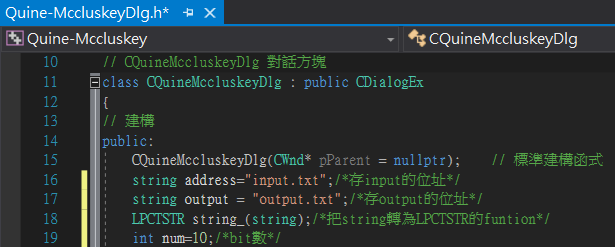
建立Quine的class來做Quine McClusky Procedure並且處理從讀檔到輸出。



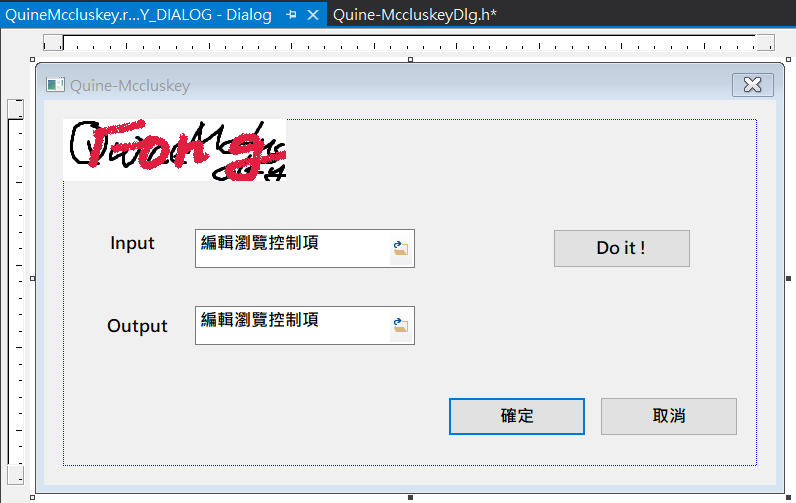
1. 讀檔:用getline一次讀進一行，利用讀進的數字宣告Minterm物件，並且藉由判斷minterm內包含1的個數區分並建立第1 round的表格(每個minterm皆為只包含一個minterm的Implicant)，存進state\_temp (Column,bit 1的個數,項目編號) ，假如讀到“()”代表為don’t care，存進儲存don’t care資訊的vector
2. 用Implicant物件的addable(Implicant&)function檢查可否結合，完成每個round相銷後檢查只被使用到一次的minterm來找essential prime implicant。
3. 將未標記的groups以及未被essential prime implicants包含的minterms分別存成新的vector，傳進Petrick(int step, vector<int>temp)這個function，用遞迴列出所有可能的組合，最後從中尋找最簡的組合。
4. 輸出結果。

MFC:

1. 在對話方塊class的標頭檔裡宣告會用到的參數



1. 設置好各項功能方塊的位置



OnBnClickedOk() OnBnClickedCancel()

OnEnChangeMfceditbrowse5()

OnEnChangeMfceditbrowse2()

OnBnClickedButton1()

1. 完成功能方塊函式內的程式碼

