311591015 葉憶嬋

Programming Assignment 1: Quine-McCluskey Method

讀檔：

1. 若輸入字串為 ".i "，則設定為 variable 個數。
2. 若輸入字串為 ".m "，在讀取到 ".d " 為止，都持續讀入數字，並將數字設定為 on set 字串。
3. 若輸入字串為 ".d "，則將讀到的數字設定為 don’t care (dc set)字串。

第一部分：定義 Quine-McCluskey 中的函式

1. 從 input 檔中讀取 variable 個數、on set、dc set 的值。
2. 從十進制轉換成二進制。接著將 on set 和 dc set 都放入 implicant 並用 vector 存起來。
3. 計算所有 implicant 中 1 的數量來分組，由小排列到大，再將不同分組的 implicant 互相合併。
4. 給定一個具有 Boolean 功能的 vector，作用是建構出 prime implicant chart。接著查看兩兩 implicant 有無 merge，若是有 merge 則在 check chart 中設定 Boolean 值為 true。
5. 根據 Boolean 值為 false 的 check chart 可以找出 prime implicant，再利用 sort 功能即可排列出對應的 prime implicant 序列。
6. 設定若 prime implicant 的 size 大於 15，則只輸出前 15 個 prime implicant。到此步驟已完成作業的前半部分。

第二部分：定義 Petrick Method 中的函式

1. 為了求得 POS，要在 prime implicant 中找出沒有跟 on set merge 過，而且不重複的項次，即為 essential prime implicant。
2. 求得 essential prime implicant 後，能得到簡化的 product of sum，再將它存入 POS 的 vector 中。
3. 將 POS 轉成 SOP，並用 sort 功能將 SOP 依照順序排列。
4. 利用 iterator 功能，搜尋不包含 " - " 的 SOP 各項次，也就是算出 0 跟 1 的總數，即為 literal 的個數。
5. 運用 min_element 選出各個解中，literal 值最小的一組解。
6. 接著輸出對應那組 literal 的 SOP 序列，即完成作業後半部。

Runtime：

1. 在使用 for 迴圈時，於 statement 判斷條件給定一個 int 值。

   例如：使用 i < vSize，而不是 i < vector_name.size( )，以減少時間。
2. 在使用 iterator 功能時，由於要搜尋 vector_name.end( ) 會比較耗費時間，所以改成先宣告好，再放入 iterator 的 statement。

   並且程式碼應該寫成++it，而不是 it++。

   例如：set<string>: : iterator beg = sop[i].begin( ) ;

   set<string>: : iterator end = sop[i].end( ) ;

   for(it=beg; it != end; ++it) {