Compiler Final Project-說明文件

原始程式 : plone.c

剖析執行檔 : compiler

測試程式 : correct.c / Error1.c / Error2.c / Error3.c

# 文法規則

1. <Program> ::= {<VarDeclaration> ';'} {"void" "FunctionName" '(' ')' <CompoundStatement>} MainFunction '(' ')' <CompoundStatement>

2. <CompoundStatement> ::= '{' { <VarDeclaration> ';'} {<Statement>} '}'

3. <VarDeclaration> ::= {<IntDeclaration> ';' | <StringDeclaration> ';'}

4. <IntDeclaration> ::= "int" [<Identifier> | <AssignmentStatement>] {',' [<Identifier> | <AssignmentStatement>]}

5. <StringDeclaration> ::= "char" <Identifier> '[' ']' '=' <String> {',' <Identifier> '[' ']' '=' <String>}

6. <Statement> ::=

[

<CompoundStatement>

|<AssignmentStatement> ';'

|<CallStatement>

|<IfStatement>

|<WhileStatement>

|<Do\_WhileStatement> ';'

|<ReadStatement> ';'

|<WriteStatement> ';'

]

7. <AssignmentStatement> ::= <Identifier> '=' <Expression>

8. <CallStatement> ::= "FunctionName" '(' ')'

9. <IfStatement> ::= "if" '(' <Condition> ')' <Statement> {"else" <Statement>}

10. <WhileStatement> ::= "while" '(' <Condition> ')' <Statement>

11. <Do\_WhileStatement>::= "do" <Statement> while '(' <Condition> ')'

12. <ReadStatement> ::= "scanf" '(' ' " ' '%' 'd' {'%' 'd'} ' " ' ',' '&' <Identifier> {',' '&' <Identifier>} ')'

13. <WriteStatement> ::= "printf" '(' ' " ' '%' \'d' | 's'\ {'%' \'d' | 's'\} ' " ' ',' <Identifier> {',' <Identifier>} ')'

14. <Condition> ::= <Expression> \ < | <= | == | != | > | >= \ <Expression>

15. <Expression> ::= [+|-] <Term> {\ + | - \ <Term>}

16. <Term> :: = <Factor> {\ \* | / \ <Factor>}

17. <Factor> ::= <Identifier> | <Number> | '(' <Expression> ')'

18. <Identifier> ::= <Alpha> {<Alpha> | <Digit>}

19. <Number> ::= <Digit> {<Digit>}

20. <Alpha> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|

a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

21. <Digit> ::= 0|1|2|3|4|5|6|7|8|9

22. <String> ::= "任何非雙引號的字元集合"

# 製作過程

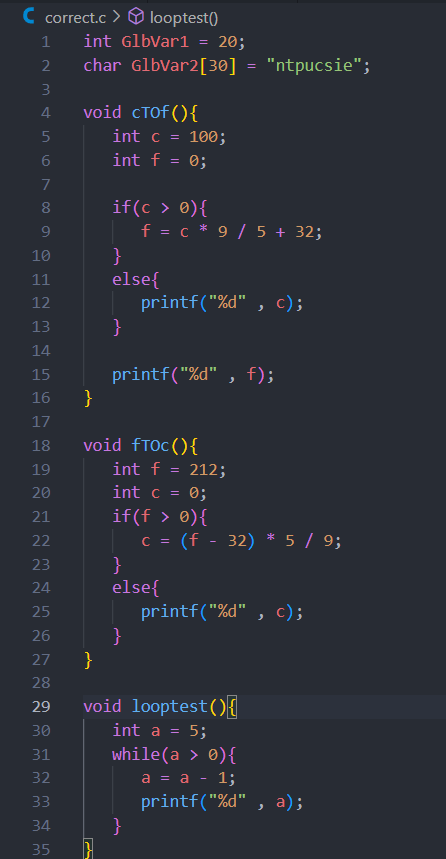
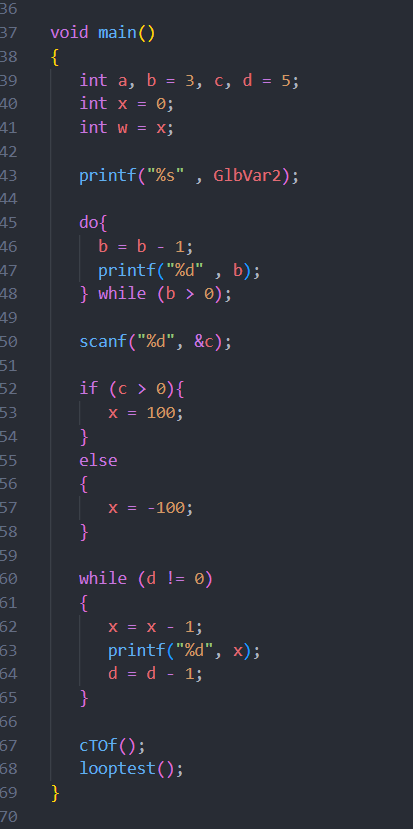
* 撰寫方式 : 使用第12章的範例程式進行修改
* 困難點 :
  1. C的語法跟plone的差異非常大，而且範例程式其實有很多地方沒有寫完整，這些都需要我們慢慢trial & error才能知道有缺哪些地方。而且裡面包含非常多 header file & source file，所以在修改code時，會有很多檔案需要一併修改內容，有時候某一個檔案忘記更動，可能需要找個20、30分鐘才找的到問題所在。
  2. C的語法提供比較完善的功能，但礙於目前技術不足，所以對於測試程式的撰寫方式較嚴格
* 解決方式 : 限制測試程式的格式

例如 (無全數列舉) :

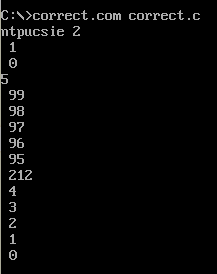
1. 不能使用variable++ / variable- -
2. 宣告僅有int和string
3. Function沒辦法傳參數

# 製作成果

## 正確程式 :



## DOS BOX 執行結果 :



cTOf執行結果

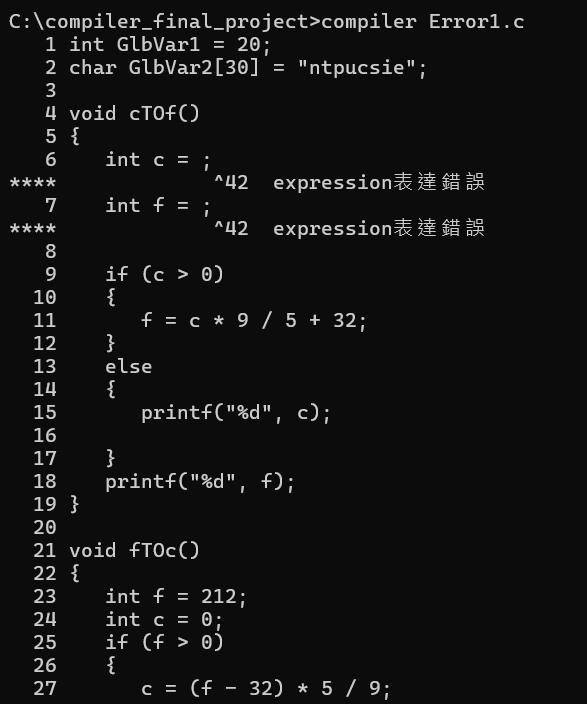
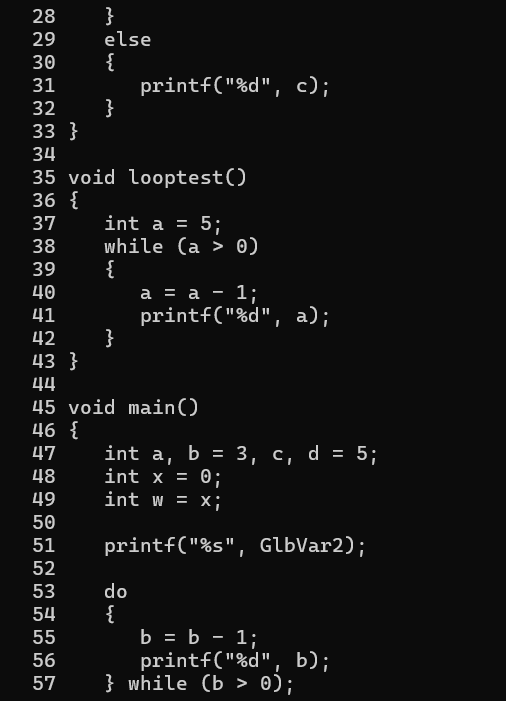
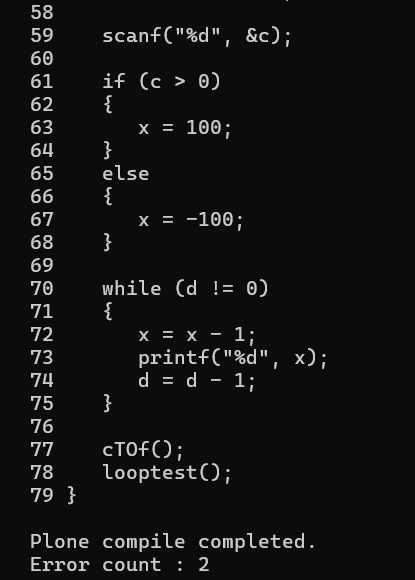
looptest執行結果

While loop五次

Scanf()

Do\_while loop三次

## 錯誤程式1 : cTOf中有assign符號卻沒有assign值

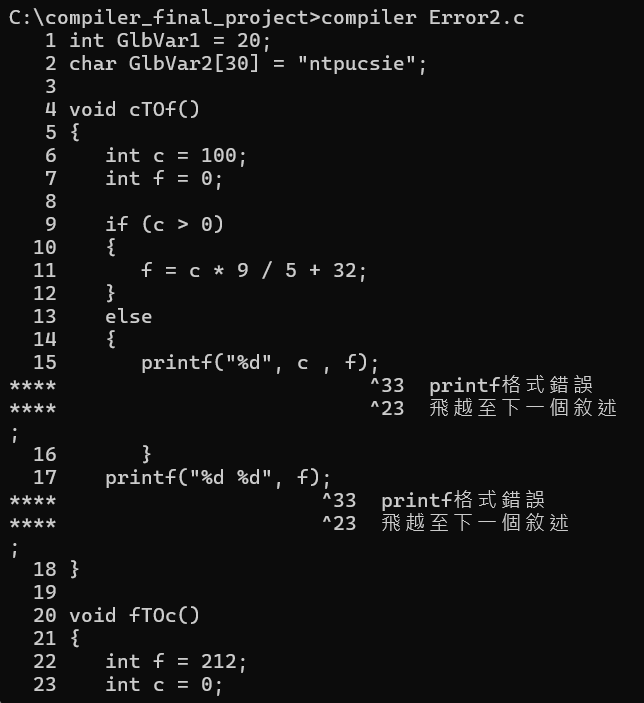
**Cmd結果 :**

## 錯誤程式2 : read/write格式錯誤(未知數數量、參數格式)

scanf格式錯誤(缺少&)

未知數數量與參數未對應

# Cmd執行結果 :



# 錯誤程式3 : 識別字範圍、屬性不符

GlbVar2是string

abcd沒有宣告過

a是在main函式宣告的

# Cmd執行結果 :

# 製作心得

一開始就有預期這個期末專案會花上很多時間去了解並製作，但在真正開始接觸後，光是熟悉、理解就花了3、4天。當然，多少存在一點拖延心態，想說時間還久就慢慢來。

從一開始撰寫文法規則，查找網路資料，中途發現自己無法從網站中擷取部分的規則出來改寫，到最後還是得仰賴chatgpt的幫助，才能將適當的文法規則建立起來。這實在是令人有點小挫折，以為上了一個學期的課程，可以不依靠工具自己一步一步實作。

有了文法規則以後，再加上前幾天對code的理解，後面的進度其實就蠻舒服的。從一開始的宣告、assign，到後面if else、while、read、write按照規則下去寫，其實都寫得蠻順利的，雖然架構寫得很醜，但基本程式都可以順利剖析。到真正要用組合語言轉執行檔時又遇到了一些小麻煩，調整組合語言、label的值也花了一兩天的時間與同學討論並撰寫，原本想加的一些功能，也因為不確定組語的寫法所以暫時沒有加上。

最後function的部分原本以為很麻煩，不過在詢問同學的意見後，加上了部分function的功能。但有許多的限制，包括只能用void宣告(因為沒做return)、不能傳參數等等的規定，再加上組合語言的改寫，終於完成這個半成品的compiler final project

寫過這個大型程式作業後，對我的邏輯思考有些許幫助，不過因為自己技術不成熟以及時間問題，有許多功能沒有好好地做完，如果之後有時間和心思，會再繼續修改裡面的架構以及補上某些功能，也許在摸索的過程中，可以慢慢理解最先撰寫compiler的人員的心情。

# 五.參考文獻

1. C/C++的格式  
   <https://learn.microsoft.com/zh-tw/cpp/cpp/?view=msvc-170>
2. 中文課本的投影片、code範例
3. <https://cs.wmich.edu/~gupta/teaching/cs4850/sumII06/The%20syntax%20of%20C%20in%20Backus-Naur%20form.htm>