Homework 2: Group Part

10910CS 546000 Software Project Management

王傳鈞(109062631)、方志強(109062804)、祝傳旻(109062634)、陳亮瑄(109062636)

**群組題—第一題 (a)**

問：請觀察資料夾Program中的兩支程式，利用開源或閉源工具，分別畫出他們的控制流程圖(control flow graph, CFG)（參考講義p.62, p.68），並附上過程截圖與工具說明。

答：以下分別以「工具說明」、「實現步驟」兩部分做詳細解說。

* **工具說明**

1. 本組採用開源的編譯器工具Low Level Virtual Machine (LLVM)。LLVM定義了一個通用程式中介表示法LLVM Intermediate Representation (LLVM IR)，它是一種類似MIPS instruction code的機器語言 (machine language)，且兼顧通用性以及對程式設計者友善性的改良版本。LLVM最重要的特色是optimizer，optimizer由許多pass組成；如果要執行程式最佳化與分析，可以撰寫自製的pass模組，以提升整體效能。
2. graphviz是一個開源軟體，可將文字圖形描述語言.dot轉換成影像檔.png和.ps。

* **實現步驟 (以**check\_triangle.c**為例)**

1. 執行環境：需具備LLVM 9.0、Clang 9.0、graphviz。
2. 利用Clang將待分析程式編譯成IR code以及具有可讀性的.ll檔。

$ clang -c -emit-llvm check\_triangle.c -o check\_triangle.bc

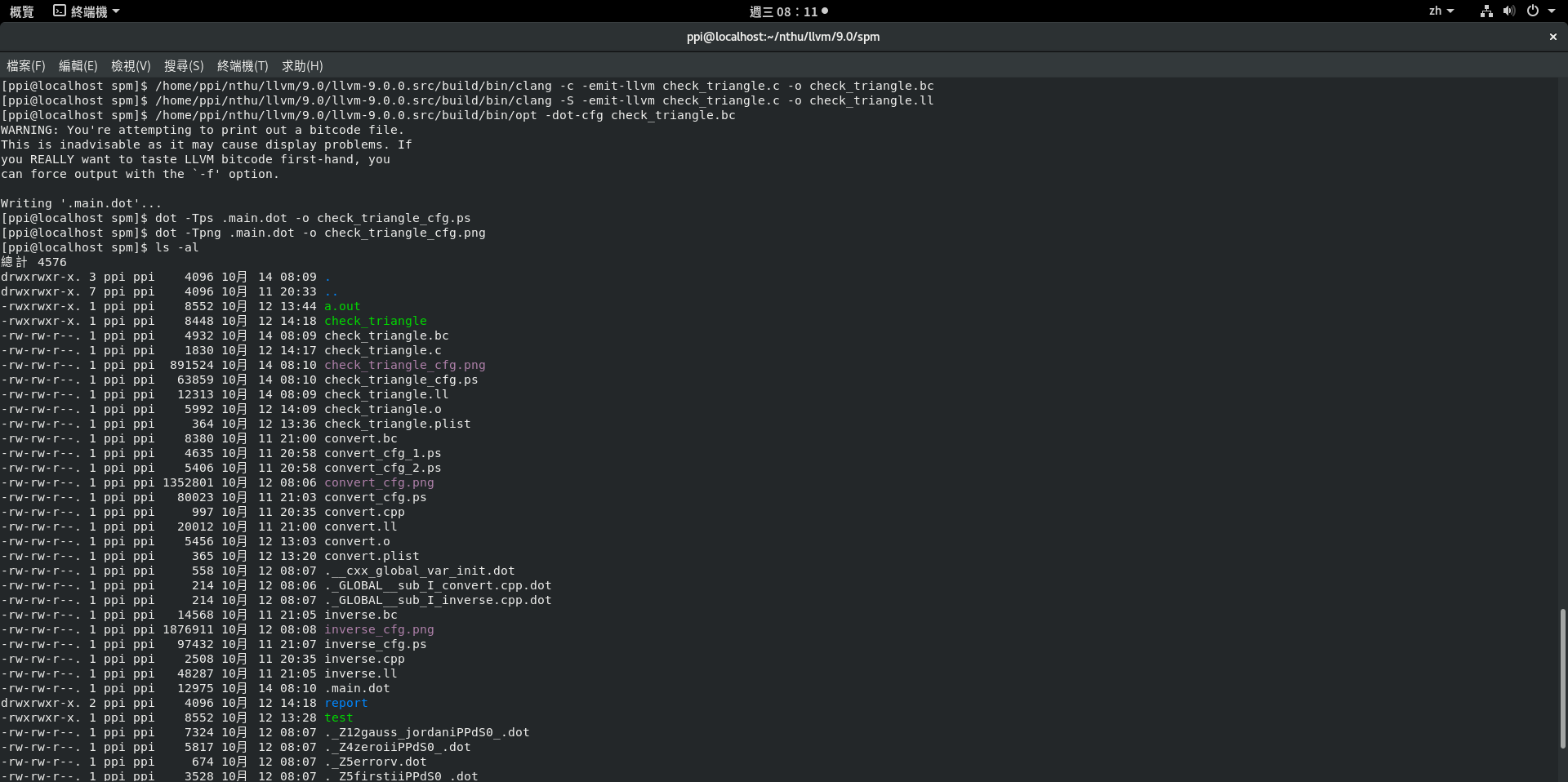
$ clang -S -emit-llvm check\_triangle.c -o check\_triangle.ll

1. 利用LLVM的opt指令產生CFG檔案，預設的輸出檔名為 .main.dot。

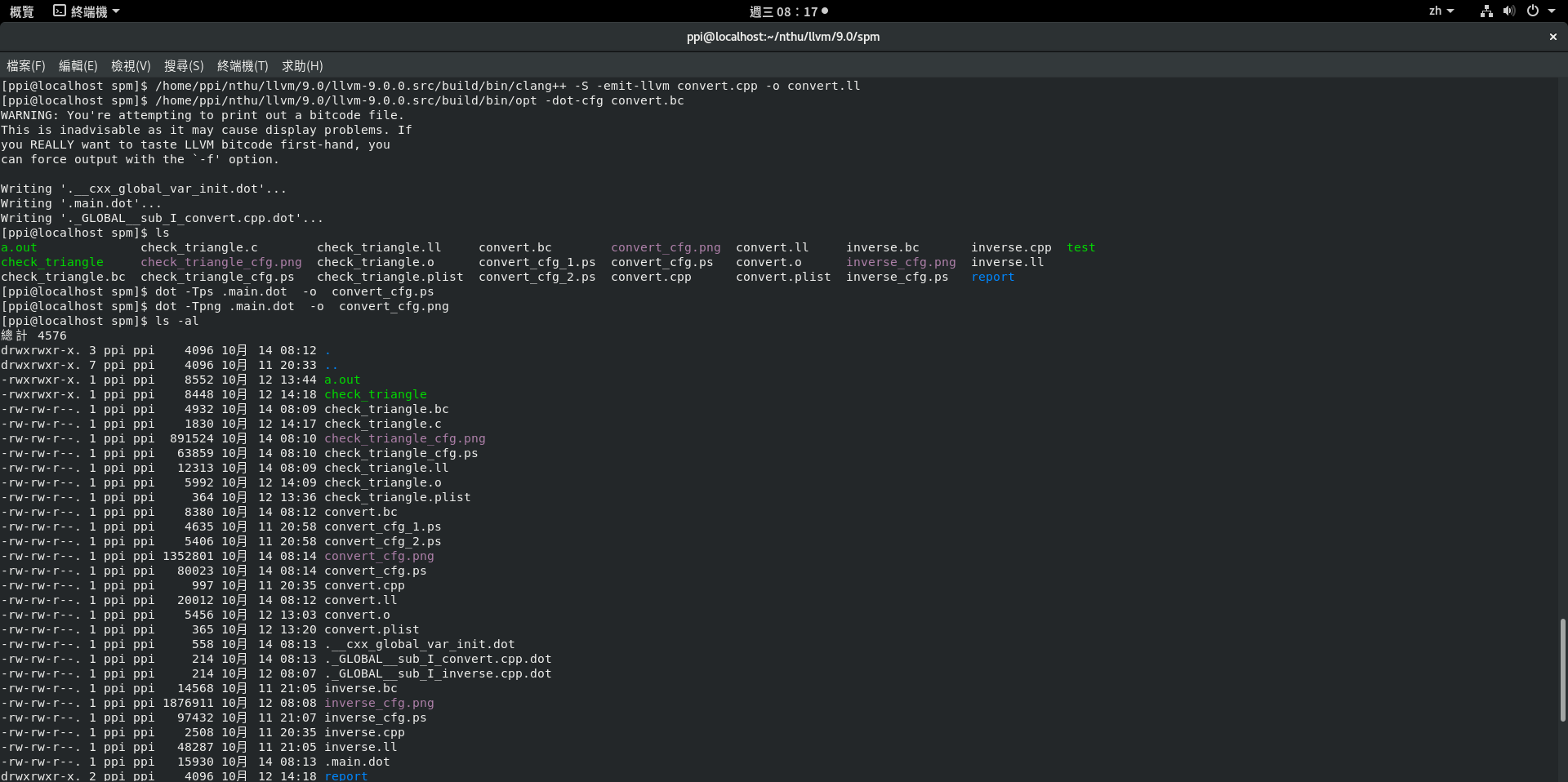
$ opt -dot-cfg check\_triangle.bc

1. 利用graphviz將前一步驟產生的 CFG檔轉換為圖檔格式 (\*.ps或\*.png)。

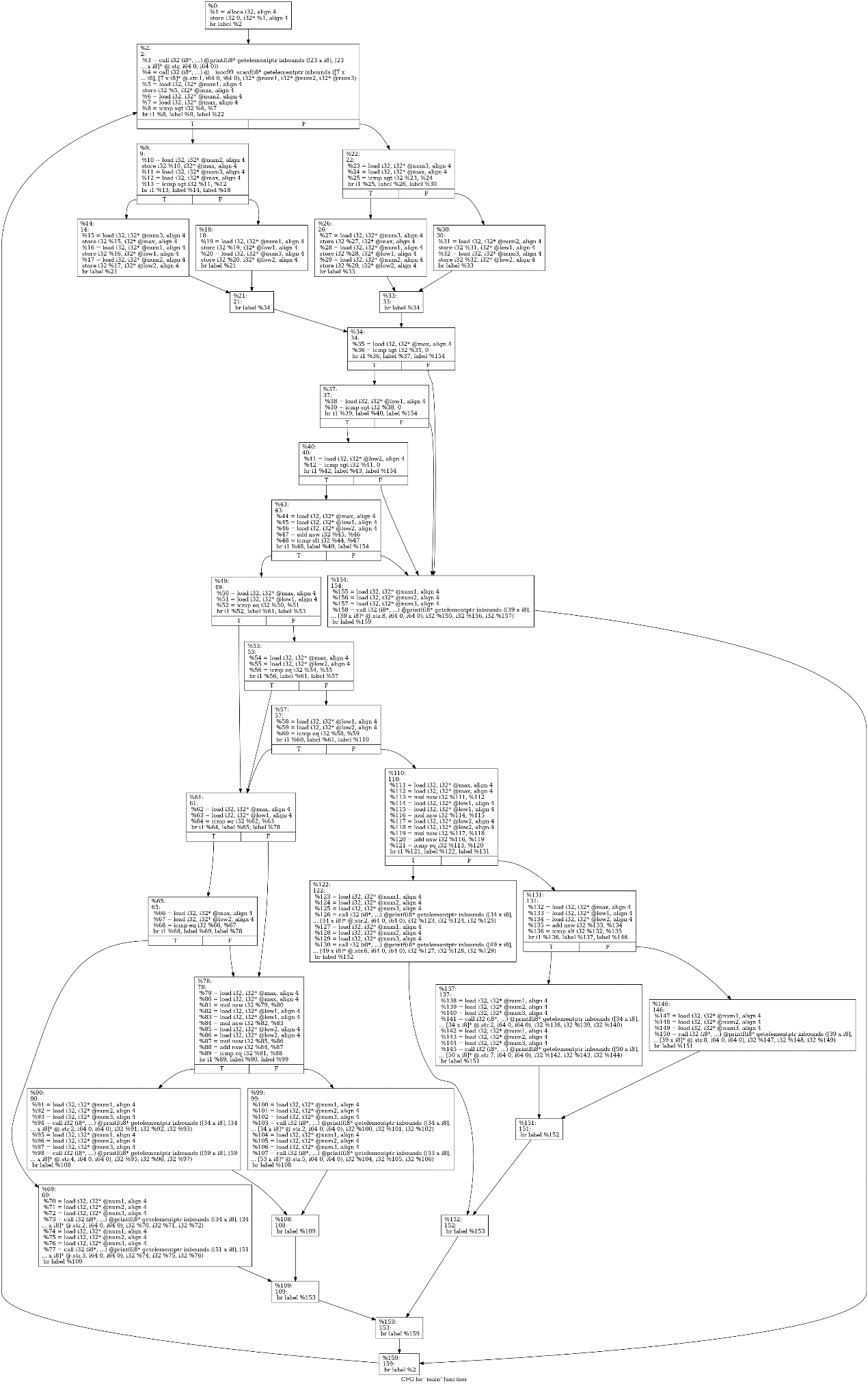
$ dot -Tps .main.dot -o check\_triangle\_cfg.ps



圖一、編譯check\_triangle.c及產生CFG檔的過程

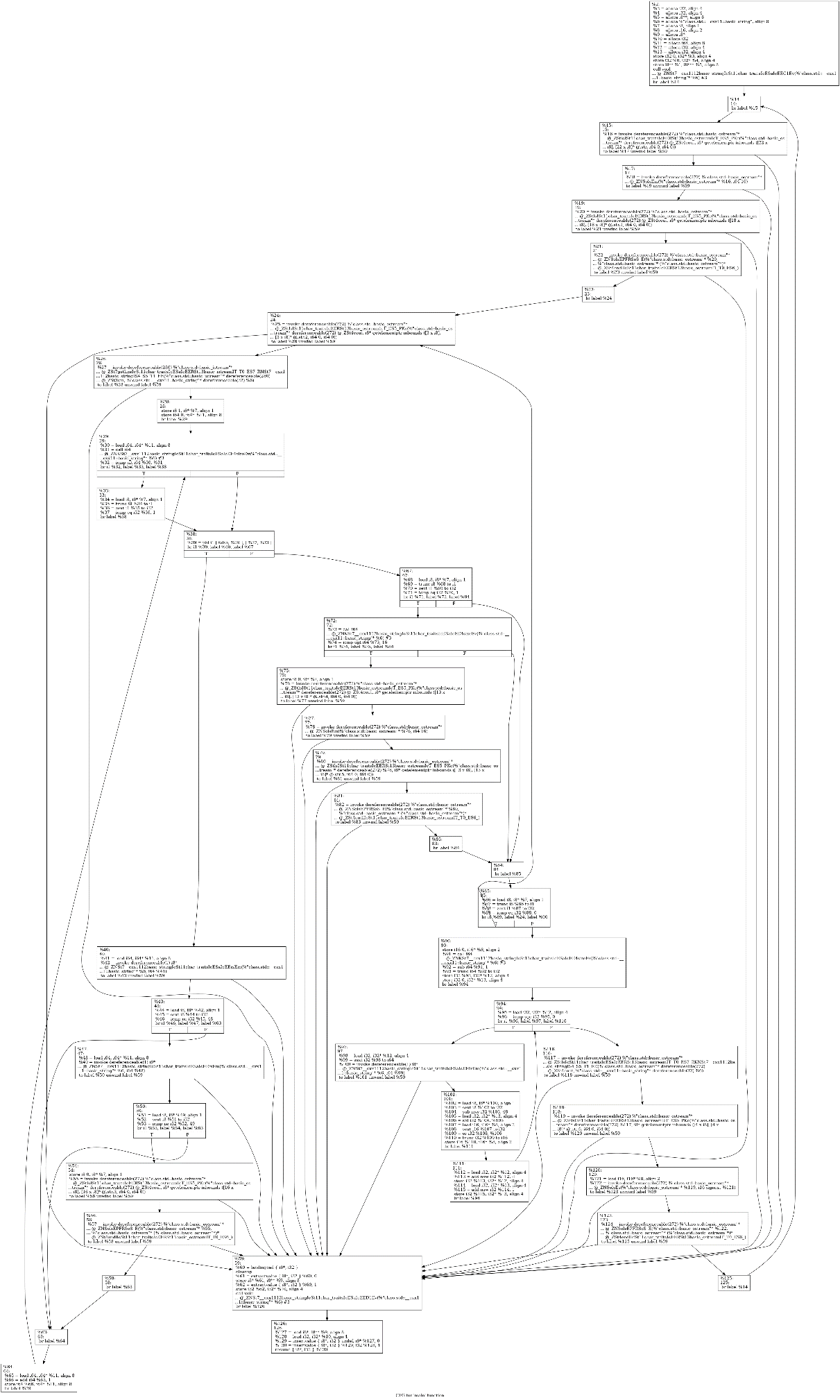


圖二、編譯convert.cpp及產生CFG檔的過程



圖三、check\_triangle.c的CFG顯示

圖四、convert.cpp的CFG顯示



**群組題—第一題 (b)**

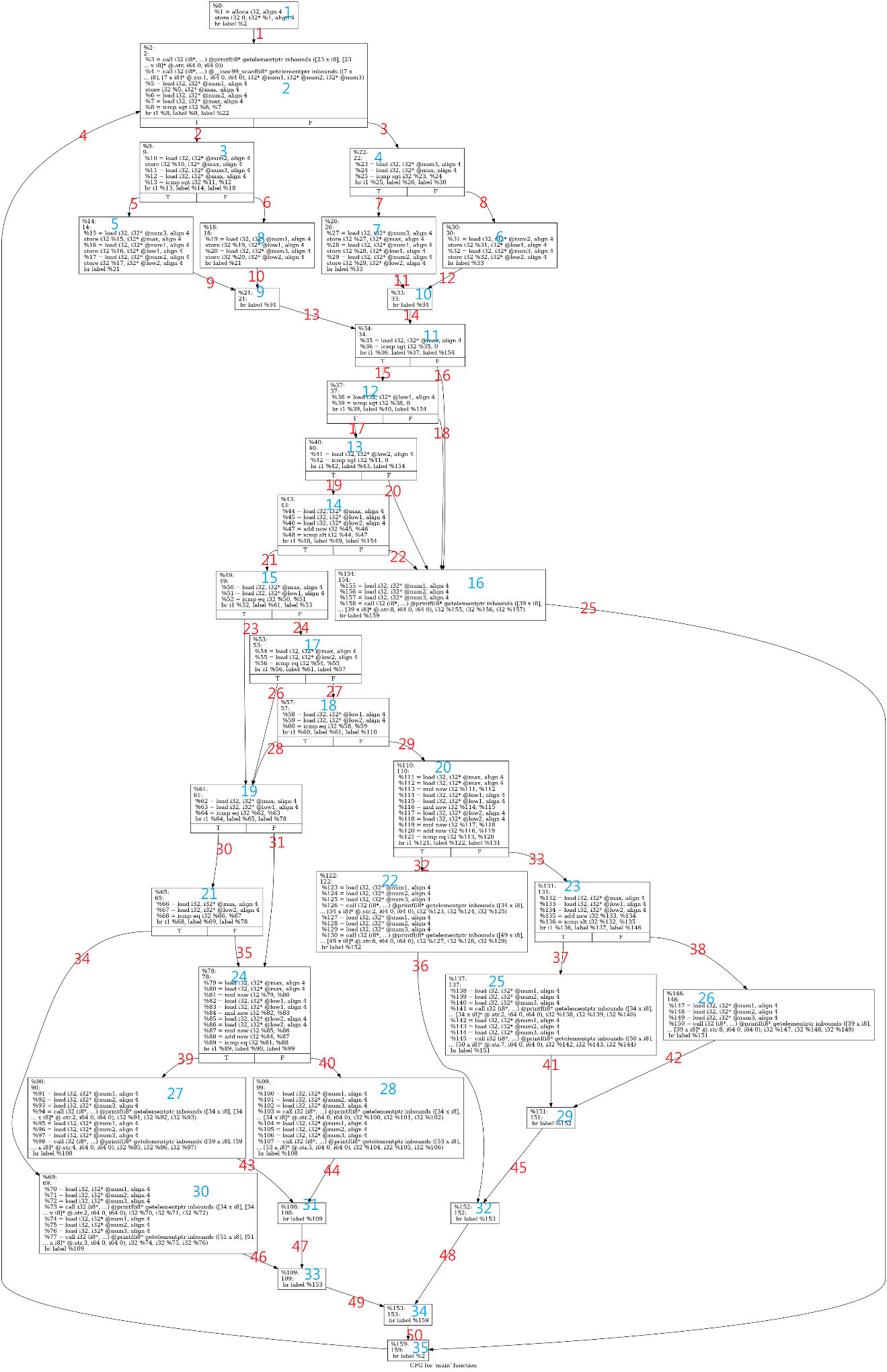
問：承上題，依據上一題得出流程圖並參考講義p.71的範例，分別計算兩支程式的圈複雜度，並列出詳細的公式（多個Module的計算方法可參考chapter 2講義p.80 Modified Cyclomatic Complexity Measures）。

答：

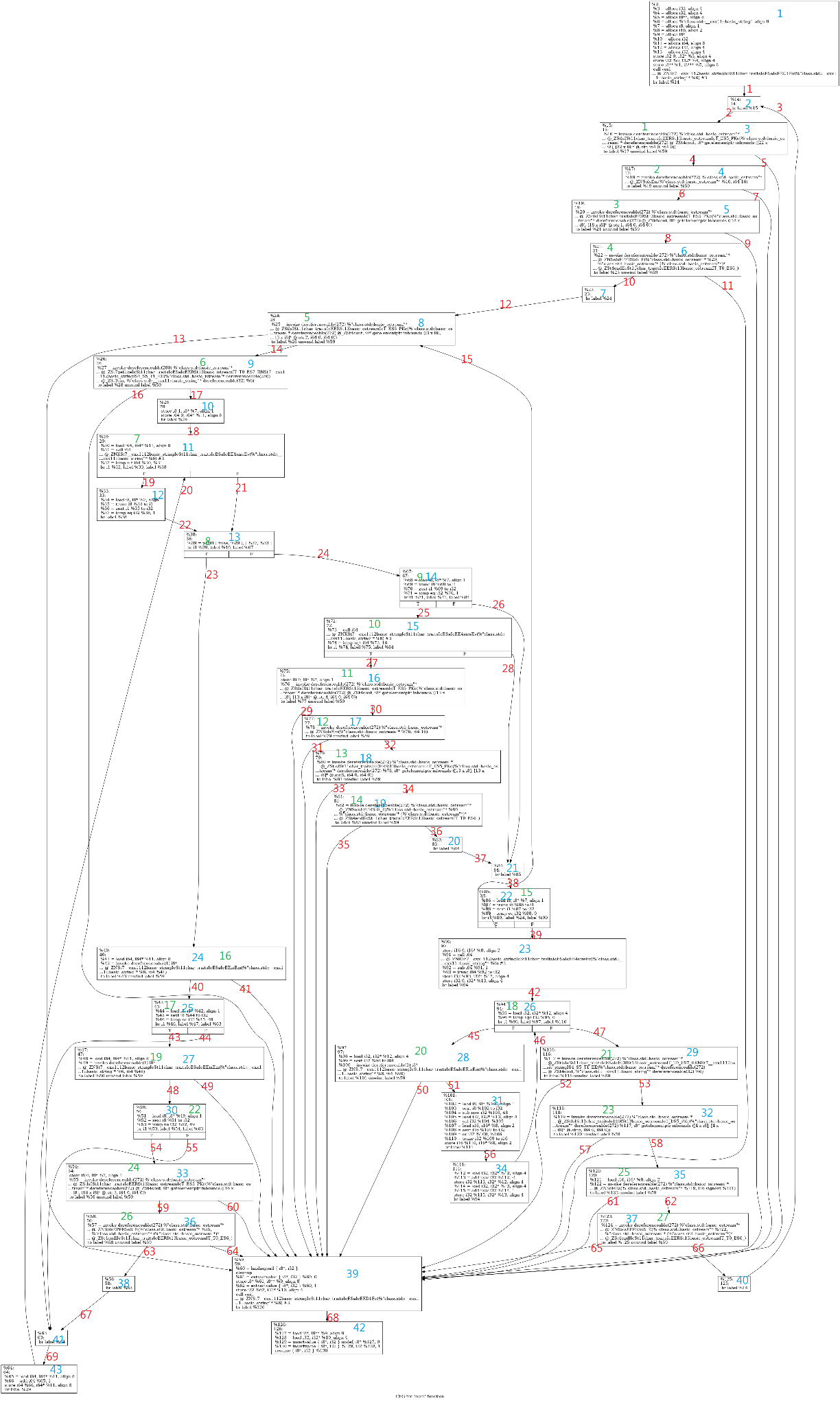
* **程式名稱：**check\_triangle.c
  + - 使用公式：
    - 、、，詳細說明可以參考圖五
    - 圈複雜度：
* **程式名稱：**convert.cpp
  + - （方法一）使用公式：
      * 、、，詳細說明可以參考圖六
      * 圈複雜度：
    - （方法二）使用公式：
      * 判定節點數 ，詳細說明可以參考圖六
      * 圈複雜度：

以上方法是透過人工分析第一題(a)所產生的CFG檔來得到各項關於圖型的特徵值，並經由 公式的計算之後所得的結果。值得注意之處：convert.cpp經由不同方法分別所求得的圈複雜度數值相吻合，但透過LLVM則會求得約莫 左右的圈複雜度。此一不容忽視之差異，我們小組認為可能來自LLVM的「optimizer」模組所致，它會分析整體程式架構進行最佳化處理，其中必定包含迴圈展開 (loop unrolling) 技術，對於大量使用到for迴圈的convert.cpp來說，此舉將大幅降低其複雜度；因此，最後導致LLVM給出的結果與人工分析不相符合之現象。

圖五、人工分析check\_triangle.c的CFG屬性



圖六、人工分析convert.cpp的CFG屬性



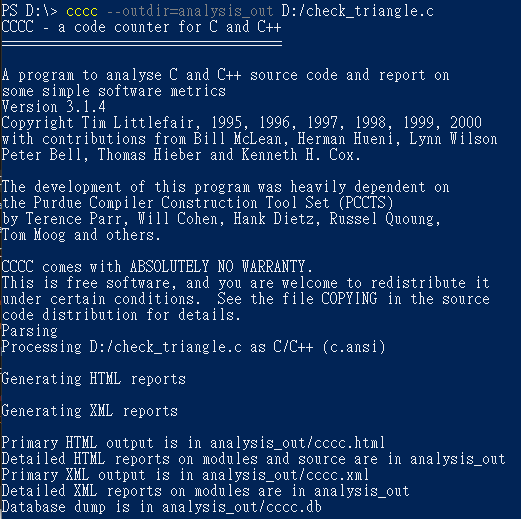
**群組題—第二題**

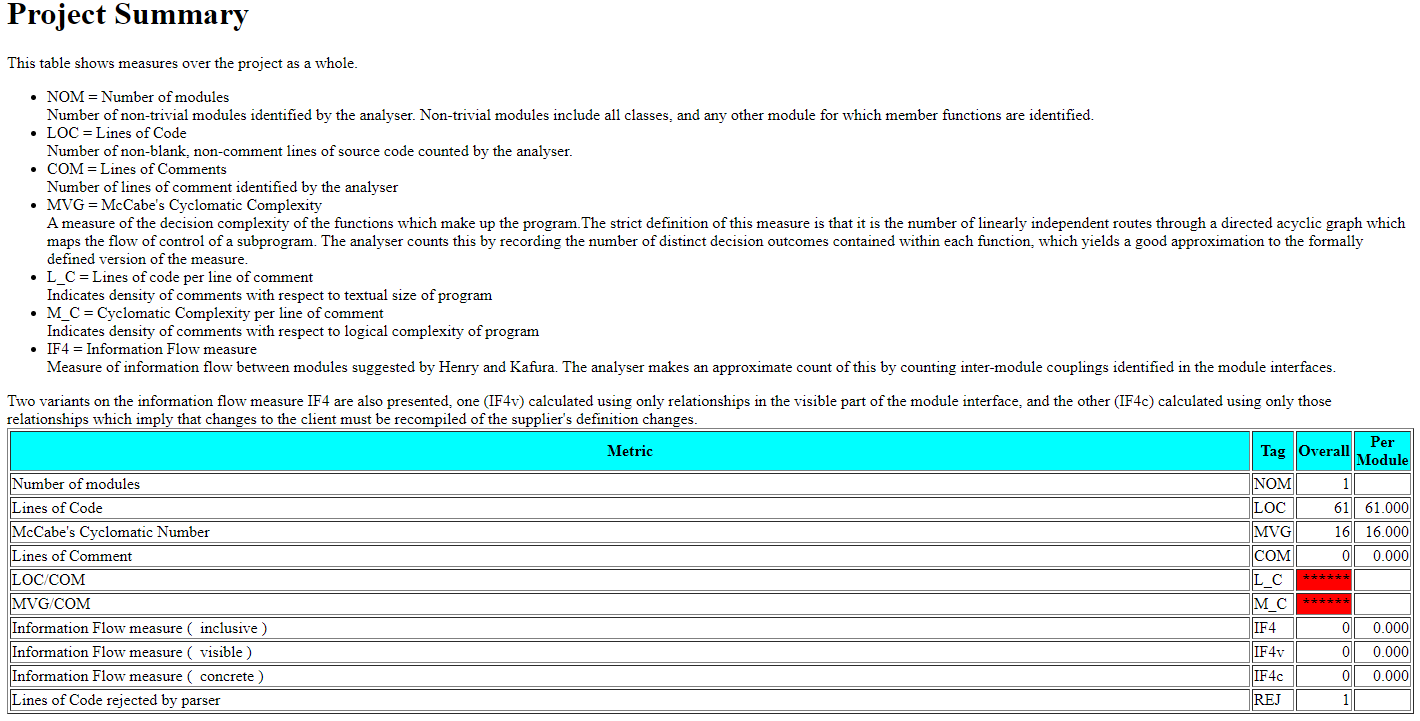
問：請同學先就作業一選擇的專案管理工具，尋找是否有軟體度量的功能，並截圖說明此功能。若作業一使用的專案軟體工具不具軟體度量的功能，則請同學另外找軟體度量的工具做說明，開源閉源工具皆可。

答：這次本組使用的軟體度量軟體為C and C++ Code Counter (CCCC)，為一個開放原始碼的軟體專案，主要用於分析C++程式的下列屬性，並以HTML形式呈現分析結果。

|  |  |
| --- | --- |
| CCCC提供的度量屬性 | |
| 縮寫 | 全稱 |
| LOC | Lines of Code |
| MVG | McCabe’s Cyclomatic Complexity |
| COM | Comment Lines |
| FO, FI etc. | Fan-out, Fan-in |
| HKS etc. | Henry-Kafura/Shepperd Measure |
| NOM | Number of Modules |
| WMC | Weighted Methods Per Class |
| REJ | Rejected Lines |

以下詳述產生CFG之流程與操作步驟：

1. 將檔案check\_triangle.c放置於目錄D:\底下
2. 終端機輸入cccc --outdir=analysis\_out D:\check\_triangle.c，就能產生以下結果
3. 開啟D:\analysis\_out/cccc.html就可檢視度量結果，如下圖所示。仔細觀察該圖片，我們就能夠輕鬆得知：check\_triangle.c具有LOC = 61和MVG = 16



**一○九學年度上學期 軟體專案管理 作業二 小組討論會議紀錄**

時間：109年10月13日（星期二）上午11時00分

地點：台達館730研究室

出席：王傳鈞、祝傳旻、陳亮瑄、方志強

記錄：王傳鈞

1. **開會事由：軟體專案管理作業二執行細項討論**
2. **討論事項**

**案由一：**個人題、群組題的分工項目

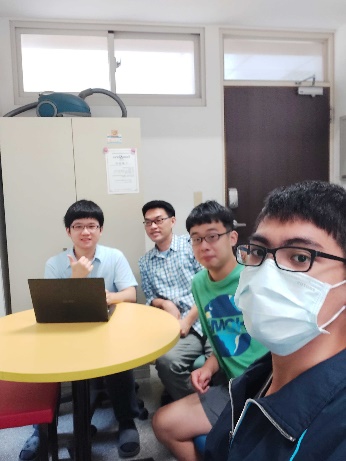
**說　明：**

根據軟體專案管理作業二之規範，共分為個人題與兩大題的群組題。為釐清群組題需由誰完成，因此需要進行各項工作的分派與認領。

**決　議：**

1. **本次作業預定於10月20日（二）收集個人題完稿、群組題草稿**
2. **繪製control flow graph (CFG)之工具為LLVM**
3. **本次作業之分工項目請見表格一**

|  |  |
| --- | --- |
| **表格一** | 分工事項 |
| 王傳鈞 | 出席人員記錄、彙整作業內容、上傳團體作業 |
| 陳亮瑄 | 群組題第一題（b） |
| 祝傳旻 | 群組題第二題 |
| 方志強 | 群組題第一題（a） |

****

1. **臨時動議：無**
2. **散會：12時00分**
3. **附錄：會議出席照片**