程式設計 作業2			
系級	航空與太空工程學系		
學號	F44134057	姓名	邱翊碩

1. 說明直譯式語言與編譯式語言的差異。(10%)

編譯式的程式語言是把程式碼從頭到尾解讀後再轉換成二進位的機器碼整個過程結束後，才開始執行這一個程式本身，如：C、C++兩者即是編譯式編譯器，在執行的效率也會比較快速。

直譯式的程式語言則是在一邊把我們打出的高階語言轉換成機器碼的同時一邊執行它，不會產生執行檔(.exe)，當碰到了錯誤的時候才會停止編譯和執行，重新執行需要整個重來，加上沒有執行檔，在效率上遜於編譯式編譯器，不過在 Debug 上是比較人性化的，在 Debug 的速度是可以比較快。如 Python、HTML、BASIC，不過也有像 Unity 同時擁有兩種特性的程式語言存在。

2. 描述 C 語言程式架構的 4 大區塊。(15%)

C 語言分為四個區塊如下：

A. 前置處理指令區

此區通常是用來宣告、調用在寫某段程式所需要用到的函式，如：include <……>，這一段程式都會寫在最前頭，從寫好的資料中找到自己所需要用的到的函式使用，有了別人所寫好的程式基礎下編寫我們想寫的程式可謂是必須的，沒有了這些基礎我們每寫一段程式都要重打一遍 printf 的函式太浪費時間和精力，對於解決問題是沒甚麼幫助的，不過對於學習的幫助是挺多的:)。

B. 全域變數及整體函式宣告區

這個地方通常介於 include 和 main 函式之間，可能會用來宣告

函式或者一些函式或者全域變數，使某一變數在各個區塊的函數都可以使用，但這就比較需要注意到記憶體方面的基本知識了。先將函式在這宣告甚至編寫也是 OK 的，因為 C 普遍上來說是從上到下閱讀的，如果我們把函式宣告放在 main 函式下方，當 main 函式需要呼叫某自訂函式時就會出現問題，因為那個自訂函式是沒還有被宣告存在的。

C. 主函式區

主函式區主要就是 main 函式裡頭的內容，程式的大綱通常都會在這裏面在寫好，如函式的呼喚等等。

D. 自訂函式區

這一區塊通常是用來撰寫與管理自訂函式的區塊，雖然函式是可以放在主程式的上方和下方，但放在上方的時候如果程式過多就可能帶來閱讀上的不便，所以先將函式宣告完成之後在於下方進行編寫也是一個不錯的選擇。

3. 變數及函式在使用前，都必須經過什麼動作。(15%)

變數在使用之前是必須先確定自己想要達成的目的，而為了達成那個目的並考量記憶體和數值範圍後我需要給我的變數哪一種資料型態才是最佳解，然後再給予變數宣告如 int(整數 integer)或 char(字元 character)等等。

函式在使用之前通常是需要導入函式庫(Library)才可以使用，不然就需要自己從零到有把它們刻印出來，除此之外還要注意函式的資料型態，如果某一個函式 F(x)的前面宣告是 void，那麼就不可以寫出像是 int x = F(5)或者類似的東西，而是直接呼叫使用它來編譯 F(x)的內容程式碼，同時不同的函式對應於不同的使用方法，在使用別人寫好的函式之前最好還是先知道呼叫它的注意事項和方法。

4. 在 C 語言中 ; 代表的意義為何？(15%)

「;」在程式語言表示的是一個程式敘述的結束，在每一段的程式後都要加「;」，不然電腦會無法讀取我們的程式，就會有 Error 產生，不過在 VS2015

中，像#include 或#Define 在前置處理區的程式就可以不需要加「；」，在其他的編譯器也有類似的情況，就多寫程式記得少加那幾個分號就可以了。

5. 說明在 C 語言中 // 與 /* */ 的差異。(15%)

//在 C 語言中是拿來做為註解的符號，在//後面的程式通常顏色呈現的方式會比較特殊，而在那之後的文字電腦是不會去把它編譯成要執行的內容的，有時候一段程式會包含著複雜的概念或者運算，而//可以用來表示這段程式的意涵讓閱讀者有較佳的閱讀體驗。而/**/傾向於更大範圍的註解或是說明，像是讓閱讀者知道這段程式的大綱之類的，在許多的程式語言中也是同樣的方式註解如 C++、Java、Python 等等。

6. 說明撰寫程式的良好習慣。(15%)

A.加上註解

程式是常常會給其他人閱讀的，而在每幾段程式後面加上適當的註解對於閱讀者是一個大大的福音，設想今天我想用一個開源的程式，打開程式碼後看得七竅生煙還是無法完全理解某段程式碼，這對於開源所想要達到的目的是相反的，同時也很折磨閱讀的人。

B.分號之後換行

如同上面提到，分號在程式語言代表的是一段程式的結束，所以每加上一次分號就換行也是寫程式中的好習慣，如果不換行的話雖然不會產生編譯錯誤，但對於閱讀、除錯等是相當不方便的，之後再次打開閱讀是很有可能自己都看不懂的，分享自己的程式碼給其他人更不用說。

C.適當的加上空格和內縮

適當的加上空格也是為了閱讀上的方便，在許多大型 Project 中會有上百上千行的程式碼，閱讀擠在一起的程式對於直接的閱讀在到思考上是相當的費神，所以有漂亮的縮排是對於他人也是對於自己的一種體諒。在 for 或者 while 之後下一行進行內縮也是，讓閱讀的人知道某段程式到底發生了甚麼，尤其是多層的巢狀迴圈。

7. 說明原始程式、目的程式與執行檔的差異。(15%)

原始程式就是我們現在所在撰寫的高階語言，是供人類所閱讀的語言，方便人類直接管理與開發，副檔名包含像是.cpp、.py，包括了英文符號與數學的運算方便人類理解。目的程式是介於原始程式和執行檔的檔案，通常是由二進位的檔案(由 1 和 0 組成)表現，不過電腦的 CPU 依舊是沒有辦法閱讀這一段檔案，所以還會經由 linker 進一步處理後產出，副檔名的表現上通常是.obj，會因為作業系統的不同而有所改變，執行檔也是如此。執行檔則是編譯式編譯器最後產出的二進位檔案，這是人類不方便所閱讀的，是交給電腦的 CPU 閱讀並執行的檔案，副檔名的表現上有像是 Window 的.exe，是我們高階語言變換到低階語言最後的產出，而三者檔案的大小也有所不同。

重點下來大致有：

a. 檔案大小的不同
b. 提供閱讀的對象不同
c. 表示的方法(如二進位)不同
d. 副檔名上的不同