

逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

網 路 教 學 網 站 -

排 序 法

學 生：駱 啟 民 (四 丙)

指 導 教 授：黃 溪 春

中 華 民 國 九 十 二 年 四 月

圖表目錄.....	3
序論:.....	4
第一章:導論.....	5
1.1 動機.....	5
1.2 目的.....	5
1.3 程式概論.....	5
第二章:開發環境及開發程式簡介.....	6
2.1 開發環境簡介.....	6
2.1.1 Java 簡介.....	6
2.2 開發軟體簡介.....	12
2.2.1 JBuilder 簡介.....	12
2.2.2 Jbuilder 的應用.....	12
2.2.3 J2SDK 簡介.....	14
2.2.4 J2SDK 的應用.....	15
2.2.5 perl 簡介.....	16
第三章 網站架構.....	17
第四章 程式實作.....	19
4.1 程式概論.....	19
4.2 程式版面的配置及按鍵實作.....	19
4.3 繪圖引擎的撰寫.....	22
4.4 排序演算法的輸入.....	25
4.4.1 Bubble Sort 演算法.....	25
4.4.2 Selection Sort 演算法.....	26
4.4.3 Insertion Sort 演算法.....	28
4.4.4 Quick Sort 演算法.....	29
4.5 測驗題實作.....	31
第五章 發生的問題及心得.....	32
5.1 發生的問題.....	32
5.2 心得.....	33
參考書目.....	33
附錄 A Bubble Sort 的 Applet 程式實作.....	33
附錄 B Selection Sort 的 Applet 程式實作.....	44
附錄 C Insertion Sort 的 Applet 程式實作.....	55
附錄 D Quick Sort 的 Applet 程式實作.....	64
附錄 E 測驗題的實作.....	76
附錄 F 解答的實作.....	81

圖表目錄

圖 2-1 Java Virtual Machine 的架構	11
圖 2-2 JBuilder 的開始畫面	12
圖 2-3 開啟檔案的畫面	13
圖 2-4 開啟 bub 檔案的結果並於任意一處輸入 test	13
圖 2-5 編譯後的畫面	15
圖 3-1 進入網站的第一個畫面	16
圖 3-2 網站架構圖	17
圖 4-1 已完成的版面	18
圖 4-2 座標示意圖	20
圖 4-3 將陣列資料畫上	22
圖 4-4 bubble sort 的指標說明	24
圖 4-5 selection sort 的指標說明	26
圖 4-6 Insertion Sort 的指標說明	27
圖 4-7 Quick Sort 的指標說明	29

序論：

近幾年來網路就像是突然冒起頭來，從高中時期一部份人所擁有的 56k 連線，到現在幾乎是家家戶戶都會擁有 ADSL 品質的連線，也因為這個原因使每個人，可以更方便的在網路上找到更多的資訊、不管在任何時間只要一上線，源源不絕的資訊就會在你面前，不斷的展示出來。

也因為這個原因，我做了這個教學網站，希望藉著網路的便利，可以讓更多的人，能夠藉由這個網站，能夠更加了解有關排序法的每個實際動作及實做方法並在最後，附上幾個小題目可以了解自己真的理解排序的動作與否。

報告的第一章為導論，主要的內容為開發的動機、目的，第二章為開發環境的介紹及開發軟體的簡介，第三章為網站的內容，實作的過程及結果，第四章為在過程中遭遇到的困難，解決的方法，還有心得。



第一章：導論

1.1 動機

一開始會想完成這個教學網站的原因，是因為在上資料結構課程時，聽著老師在台上說著每個排序法的演算法，然後又是舉實例說明，可是聽的還是不太理解，每個動作間的過程，大概最後的印象都是，一筆資料進去，然後就變成了一筆由小到大的資料送出來，寫作業程式的時候也是演算法改一改，沒有很仔細的去理解每個動作的關係，那個時候就在想，如果可以用圖形的方式表達每筆資料的比較，和交換動作，會不會更容易理解排序的過程，在這個想法下，我用 Java Applet、Front Page 及 Perl，寫下了這個網站。

1.2 目的

從小到大我都相信，單純的藉由文字的表達，其實並不容易讓人理解一件事情，如果能夠輔助上圖形的幫助，看的到實際的動作，能夠讓理解一件事情的速度增加。所以本專題的目的是希望以圖形化的方式，來幫助學習，排序的過程，加深對於排序時每個動作的印象。在最後提出幾個小題目來測驗對於排序的理解度。

另外，我也附上了本次專題有關 Java Applet 程式寫作的程式碼，若有興趣的人，也可以下載來研究看看。

1.3 程式概論

關於本網站的內容，包含有基本的排序介紹，演算法，排序法的特點，排序的過程。為了避免使用上的不方便，加設了 J2SDK1.42 的下載教學，以避免使用者在使用上，因為版本的問題導致排序過程無法瀏覽的問題，測驗是利用 perl 所寫的 CGI 網頁，目前有 14 道題目，藉著隨機取樣的方法，隨意抽出四題作為練習，以加深對於排序法的概念。

第二章：開發環境及開發程式簡介

2.1 開發環境簡介

我所選擇用來完成這次專題的程式語言是 Java Applet，之所以會選擇這個語言的原因是因為，雖然對這套語言不熟，但也有曾經用她完成過一次的作業，在當時對這套語言，有了很深刻的印象。剛好這次的專題可以使用上，就決定選擇她了。

2.1.1 Java 簡介

JAVA 是由美國昇陽公司(Sun Microsystems, Inc.)所開發出來的一個程式語言，中文譯名為爪哇。JAVA 本身是一種物件導向(Object-Oriented)的程式語言，所以在使用學習上並不會很困難，如果您有使用過 C++語言的話，那麼寫 Java 就更能駕輕就熟了，因為基本上 Java 的使用是比 C++還要簡單一些。它的目標是以能在任何各種各樣的不同機器，不同種類的作業系統平台中的網路上，發展你所要開發的程式。也就是說不論你使用什麼機器、什麼作業系統，只要你的瀏覽器支援 JAVA，你就可以使用其程式，這伴隨瀏覽器使用的網頁程式，我們稱為 Applet。另外，除了網頁上的 Applet，JAVA 也具有發展大型應用程式的能力，並且同樣地可以跨躍不同種機器、不同種的作業系統平台的限制，可在各平台中執行。並且能在國際網際網路(Internet)上面具有「硬體/軟體中立性」互動能力的程式語言。

JAVA 寫出來的程式可分為兩類，分別是 Java Applets 與一般的 Application，而 Application 這一類就與一般的程式如 C++的作用是比較類似的，反正就是一個獨立可執行的應用程式，像 HotJava 是一個瀏覽器，且就是使用 Java 程式所發展出來的。而 Java Applets 它也算是一個獨立的程式，但它的限制就是只能在瀏覽器中來執行，原因就是因為 Java Applets 本身是 ByteCode，必須透過瀏覽器中的 Java 直譯器來解讀它，才可以被執行。

在網路上，您可以連線到美國昇陽公司(Sun)的 Java 網站，網址是 <http://java.sun.com/>，或是 <http://www.javasoft.com/>。在那

裡幾乎可以找到您所需要的 JAVA 資訊，但是語文多少是一個障礙，至少對某些人而言；沒關係，目前國內已有很多個網站提供中文 JAVA 資訊，爾後應會有更多的網站提供相關資訊。總而言之美國昇陽公司 (Sun) 是要 Java 成為一個簡單 (Simple)、物件導向的 (Object Oriented)、分散式的 (Distributed)、解譯的 (Interpreted)、強韌性的 (Robust)、安全性的 (Secure)、架構中立性的 (Architecture Neutral)、可攜帶性的 (Portable)、高效能的 (High Performance)、多重線串 (Multithreaded)、動態性的 (Dynamic) 的程式語言

1. 簡單 (Simple) :

容易撰寫程式，不需要長時間的訓練，而能滿足現代的需求。程式小型亦是簡單的一種特性，使得軟體能夠在小型機器上執行，基本的解譯器約為 40 k，若加上基本的程式庫，約為 215k。

2. 物件導向的 (Object-Oriented) :

物件導的設計是一種重心在資料和介面的技巧。若以木工為比喻，一個以物件為導向的木工，他(她)最主要的重點是即將要做的木椅子，其次才是所需要的工具；反之；一個以非物件為導向的木工，他(她)所關心的只是工具。最近的隨插即用 (Plug and Play) 亦是物件導向設計的重點。

1. 分散式的 (Distributed) :

Java 有一個很週全的程式庫，且很容易地與 HTTP 和 FTP 等 TCP/IP 通訊協定相配合。Java 應用程式 (Applications) 能在網路上開啟及連結使用物件，就如同透過 URLs 連結使用一個區域檔案系統 (Local File System)。

4. 強韌性的 (Robust) :

由 Java 所撰寫出的程式能在多種情況下執行而具有其穩定性。Java 與 C/C++ 最大不同點是 Java 有一個指標器模型 (Pointer Model)

來排除記憶體被蓋寫(Overwriting Memory)和資料毀損(Corrupting Data)的可能性。

5. 安全性的(Secure) :

Java 是被設計用於網路及分散性的環境中，安全性自必是一個很重要的考慮。Java 擁有數個階層的互鎖(Interlocking)保護措施，能有效地防止病毒的侵入和破壞行為的發生。

6. 架構中立性的(Architecture Neutral) :

一般而言，網路是由很多不同機型的機器所組合而成的，CPU 和作業系統架構均有所不同；因此，如何使一個應用程式可以在每一種機器上執行，是一個難題。所幸，Java 的編譯器產生一種結構中立物件的檔案格式(Object File Format)；這使得編譯碼得以在很多種處理器中執行。

7. 可攜帶性的(Portable) :

原始資料型式的大小是被指定的，例如"float"一直是表示一個 32 位元 IEEE 754 浮點運算數字，因絕大多數的 CPU 都具有此一共同的特徵。程式庫屬於系統的一部份，它定義了一些可攜帶的介面，Java 本身具備有很好的可攜帶性。

8. 解譯的(Interpreted) :

Java 解譯器能直接地在任何機器上執行 Java 位元碼(Bytecodes)，因此在進程式連結時，時間的節省，這對於縮短程式的開發過程，有莫大的助益。

9. 高效能的(High Performance) :

Java 位元碼迅速地能被轉換成機械碼(Machine Code)，從位元碼轉換到機械碼的效能幾乎與 C 與 C++沒有分別。

10. 多重線串的 (Multi threaded) :

Java 語言具有多重線串的功能，這對於互動回應能力及即時執行行為是有助益的。

11. 動態的 (Dynamic) :

Java 比 C 或 C++ 語言更具有動態性，更能因應時刻在變的環境，Java 不會因程式庫的更新，而必須重新編譯程式。

而「Write once, Run anywhere」是 Java 語言的主要訴求點之一，達到這個目標的主要功臣則非 JVM (Java Virtual Machine, Java 虛擬機器) 莫屬。Java 的原始程式透過 Java 編譯器編譯成 bytecode 之後，便可以在任何有 JVM 的平台上被行 不論是 Windows 還是 Unix。

一般來講，Java 的開發流程為：

1. 利用文書處理器撰寫 Java 的原始碼。
2. 經由 Java 編譯器編譯成 Bytecode
3. 可由本地端或網路端由 class loader 載入，並由 bytecode verifier 檢查。
4. 由直譯程式解譯 bytecode 或即時編譯程式 (Just-In-Time Compiler), 將 bytecode 指令轉化為原生機器碼來執行。此部分即為 JVM 的主體。

JVM 是一個虛擬機器，而 bytecode 可以看成是這個虛擬機器的機器語言。bytecode 總共定義了 202 個指令，這些指令範圍相當廣泛，除了包含資料搬移、數學運算、流程控制等這些跟一般處理器相似的指令之外，還定義了跟物件 (Object)、例外 (Exception) 以及執行緒 (Thread) 相關的一些指令，這使得 bytecode 指令集看起來相當地高階。在 JVM 中，有個 bytecode 直譯器 (Bytecode Interpreter) 來負責這些指令的執行工作。

JVM 採用 stack-based 的設計，幾乎所有指令的動作都需透過 stack 來完成，這樣的設計使得 JVM 可以不受到暫存器數目或格式的影響。基本上 JVM 對暫存器的要求只需一個 Program Counter 來

記錄執行順序，另外加上幾個暫存器來控制 stack 即可（如 vars, optop 等）。JVM 允許同時有多條執行緒運作，每一條執行緒都會分配到一個私有 Java 堆疊（Java Stack），用來放置執行時所產生的 frames。每當執行緒呼叫到一個新的 method 時，便會在 Java 堆疊內產生一個 frame 結構。每一個 frame 包含三個區塊：

局部變數區（Local Variables）

用來放置 method 所用到的所有局部變數。

執行環境區（Execution Environment）

主要用來放置 JVM 所需的一些控制資訊，如 frame 間的連結、除錯資訊等等。這個區域是 Implementation-dependent 的，不同的 JVM 可以有不同的設計。

運算元堆疊（Operand Stack）

用來暫時放置 JVM 運算過程中所需要或產生的任何資料。

除了私有的 Java 堆疊外，所有的執行緒還共用 JVM 中的兩個資料結構，一個是 Heap 另一個是 Method Area。Heap 是用來放置物件，每個物件經過 new 指令誕生後都存在於 Heap 中，並由 JVM 中的 Garbage Collector 來負責回收已經不用的物件，使用者不必自行作記憶體回收處理。而 Method Area 則相當於傳統的 code segment，主要存放著類別及類別中 method 的 bytecode

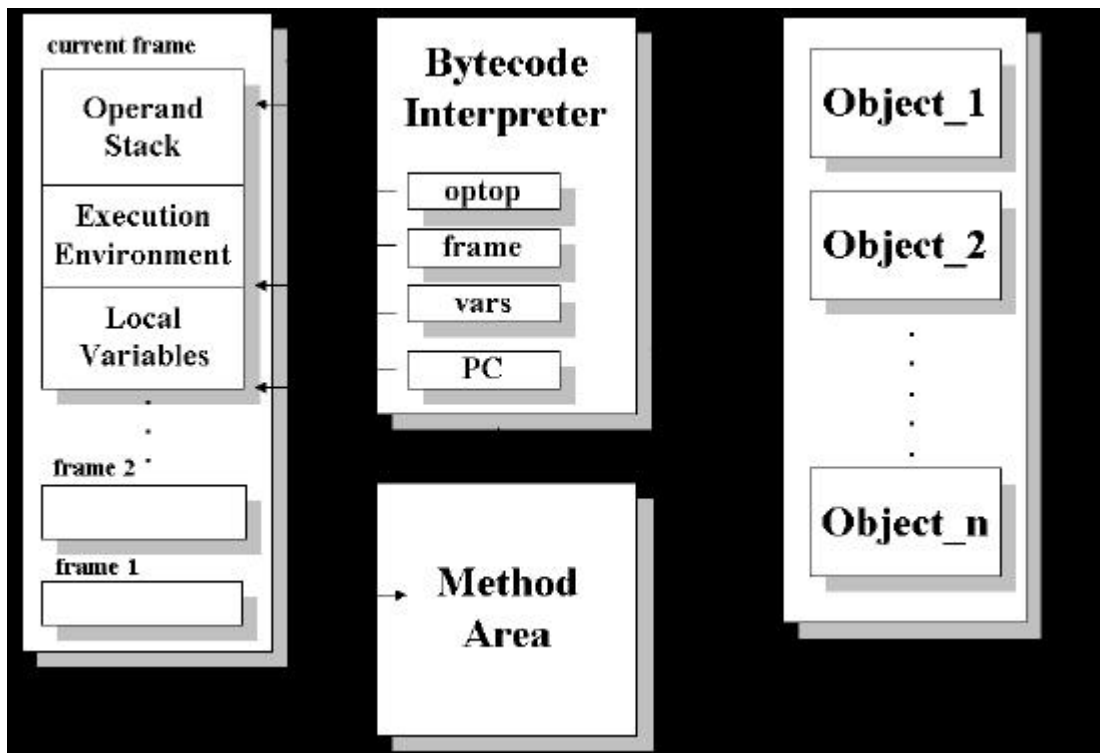


圖 2-1 Java Virtual Machine 的架構

圖 2-1 為 JVM 架構圖。模擬上圖的架構, JVM 便可以開始執行 Java 的程式。不過在 JVM 執行 Java bytecode 程式時, 有可能會需要有關 I/O、Networking 以及 Multithreading 等功能, 這部份 JVM 並未定義在其規格內, 所以在 JVM 的最底層, 還有一個 Runtime Support 的

部份, 用來向底層作業系統取得上述的 I/O、Networking 等服務。

執行 bytecode 的最基本方式為直譯式, 它的運作方式如下:

```
while (there are bytecodes) // 迴圈
{
    fetch a Bytecode // 擷取指令
    Interpret and Execute it! // 解譯並執行該指令
};
```

即是針對每一個指令有一組對應的敘述, 來模擬此指令的動作, 並反應在上圖之資料結構中。JVM 中每一個指令所對應的動作, 和其相對應的直譯程式去模擬的演算法

2.2 開發軟體簡介

一般的 Java 程式都是利用文字編輯器直接輸入 Code，等全部輸入完之後，在藉由 Java 的編譯器來決定是否有發生錯誤，但這次由同學手中取得 JBuilder，意外的發現其實在使用上是很方便的工具，雖然這次的專題，並沒有使用到太多她的功能，但不可否認的是，這絕對是開發 Java 程式的好工具，另外我們還需要 J2SDK 來幫助我們將

2.2.1 JBuilder 簡介

Borland Jbuilder，為 Borland 在 Java 開發平台上所提供的一套強大的軟體她整合了許多的功能在裡面，可是我這次只有使用到她的文法檢查功能。

2.2.2 Jbuilder 的應用

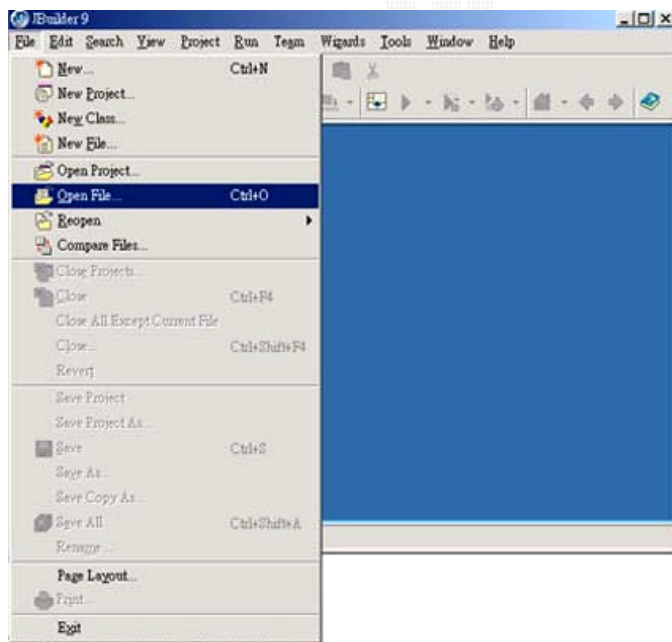


圖 2-2 JBuilder 的開始畫面

在這裡由於我只需要使用到文法檢查的功能，所以直接以 Open file 的方式來開啟已經編輯過的 .java 檔如圖 2-2 所示

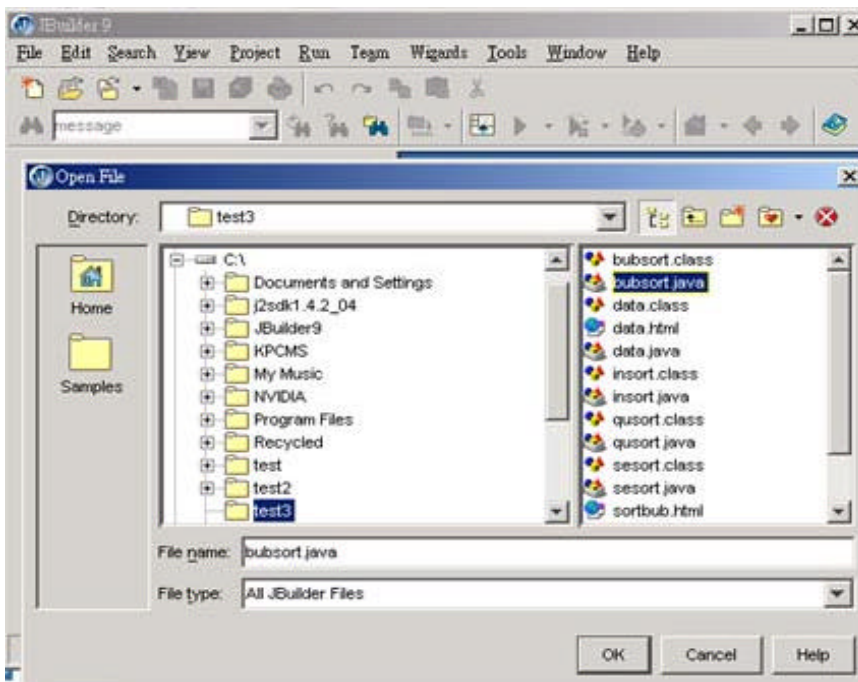


圖 2-3 開啟檔案的畫面

如圖 2-3 我們可以看到 在 test3 資料夾裡有很多個檔案現在我們點選 bubsort.java 這個檔案來作個示範

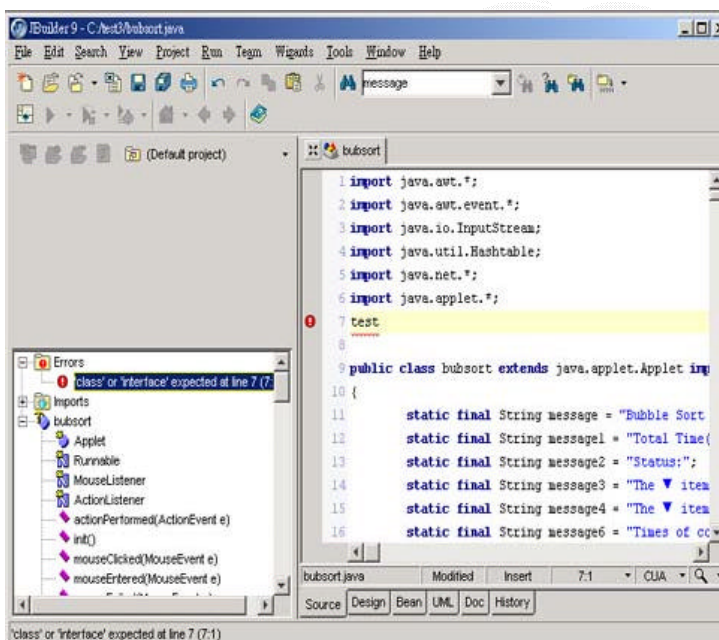


圖 2-4 開啟 bubsort 檔案的結果 並於任意一處輸入 test

由圖 2-4 我們可以看到，Jbuilder 將 test 這四個字母標上了記號，並在左方的視窗中，說明了文法錯誤及所在位置，就這點，對我來說已經是很大的幫助了，因為在一堆迴圈當中很容易就會誤打了幾個字，或是不小心將保留字拿來當作變數，抑或是不小心少了個”}”之類的 這些都可以直接提醒我們，另外它還提供了保留舊檔的功能，有時候一不小心很有可能發生砍錯檔案的意外，這個時候就可以去存檔案的資料夾中將之前的舊檔案找出來重新修改 算是一個還蠻方便的功能

2.2.3 J2SDK 簡介

另外當程式完成時就需要 J2 SDK 來幫助我們將.java 檔，翻議成.class 本次程式所用的 SDK 版本為 1.4.2。

Java 2 SDK 是使用 Java 程式語言來建立應用程式、Applet 和元件的開發環境。Java 2 SDK 所包含的工具可用來開發和測試以 Java 程式語言所撰寫，而且在 Java 平台上執行的程式。這些工具是設計在指令行中使用。

Java 2 SDK 的內容包含有

開發工具（在 bin 子目錄中。）一些工具、公用程式，可協助您開發、執行、除錯和記錄以 Java 程式語言撰寫的程式。

執行環境（在 jre 子目錄。）供 SDK 使用的 Java 2 Runtime Environment 實作。執行環境包含了 Java 虛擬機器、類程式庫以及其他支援以 Java 程式語言撰寫的程式可執行檔。

其他程式庫（在 lib 子目錄。）開發工具所需的其他類程式庫和支援檔案。

示範 Applet 和應用程式（在 demo 子目錄。）Java 平台程式設計的範例及其原始碼。這包含了使用 Swing 其他 Java 基礎類和「Java 平台除錯程式架構」的範例。

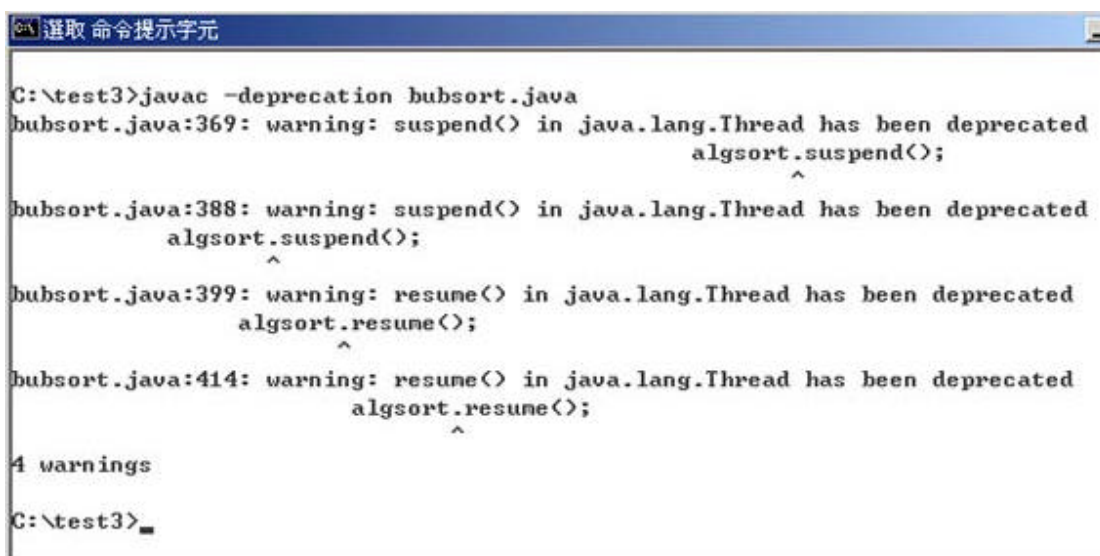
C 標頭檔（在 include 子目錄。）使用 Java 原生介面、Java 虛擬機器除錯程式介面、Java 虛擬機器設定檔介面和其他 Java 2 平台功能來支援原生程式碼設計的標頭檔。

原始碼（在 src.jar 保存檔中。）作為 Java 2 核心 API 的

所有類原始 Java 程式語言檔 (也就是 java.*、javax.* 和 org.omg.* 套件而不包含 com.sun.* 套件的原始檔)。

2.2.4 J2SDK 的應用

Java 語言的編譯器為 javac , 直譯器為 java 。由於我們寫的是 Applet 程式 必須配合 HTML 使用所以 今天我們只需要 javac 這個編譯器就夠了



```
C:\test3>javac -deprecation hbsort.java
hbsort.java:369: warning: suspend() in java.lang.Thread has been deprecated
        algsort.suspend();
                        ^
hbsort.java:388: warning: suspend() in java.lang.Thread has been deprecated
        algsort.suspend();
                        ^
hbsort.java:399: warning: resume() in java.lang.Thread has been deprecated
        algsort.resume();
                        ^
hbsort.java:414: warning: resume() in java.lang.Thread has been deprecated
        algsort.resume();
                        ^
4 warnings
C:\test3>_
```

圖 2-5 編譯後的畫面

由圖 2-5 我們可以看到經過編譯後是否有發生錯誤，若有編譯器也會將錯誤所在的地方，及錯誤的原因列出，如果沒有則成功產生.class 檔，再來只要將 class 檔寫入 HTML 文件中即可在網頁上顯示 applet 程式

```
<APPLET CODE= " sort.class " width=250 height=90>
</applet>
```

其中<APPLET>標籤表示用來標示放置網頁上的 Java applet 程式，CODE 表示 APPLETT 程式的類別名稱檔案 width、height 代表 applet 視窗的寬和高。

2.2.5 perl 簡介

Perl 是 Practical Extraction and Report Language 的縮寫，是由 Larry Wall 融合了 awk/sed/cut/grep 等 unix 上常見的工具與 C language 的語法所創造的一種 language。

Perl 是一種解譯式的語言，在速度上並沒有 C 這類編譯式的語言來得快，不過 Perl 在執行時事實上是先將原始檔即時編譯後再執行，所以速度其實並不慢。

Perl 極為適合處理用來處理一些用 C 處理並不方便，用 shell 處理又嫌太複雜的工作。Perl 強大的文字處理能力，使得它極為適合作為 CGI 程式所用的 language，這次在測驗的網頁中，就是結合了 html 語法作表單，與 perl 程式傳遞數值所完成的 CGI 網頁



第三章 網站架構

當我們進到本網站時看到的第一個畫面應該是圖 3-1

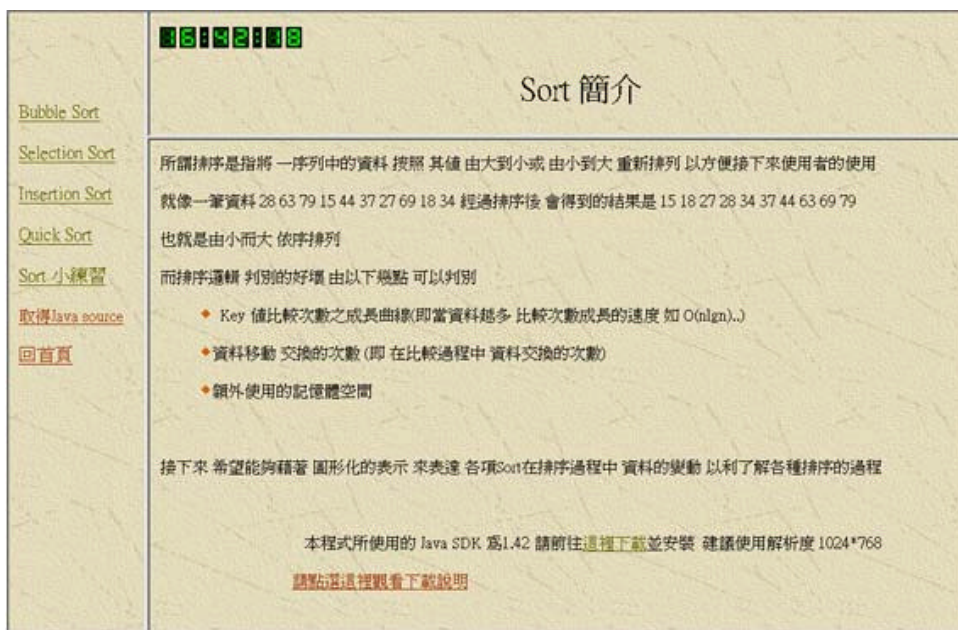


圖 3-1 進入網站的第一個畫面

在這裡，我們可以看到，左方的功能列表中可以點選所想選用的功能，在依照所點選的功能，進入不同的頁面，進行所選取的工作，其中取得 Java source、回首頁，這兩項功能是為了方便使用者所設定的，而 Bubble Sort、Selection Sort、Insertion Sort、Quick Sort、Sort 小練習，這五項功能才是本網站的教學內容。圖 3-2 為本網站的架構圖。

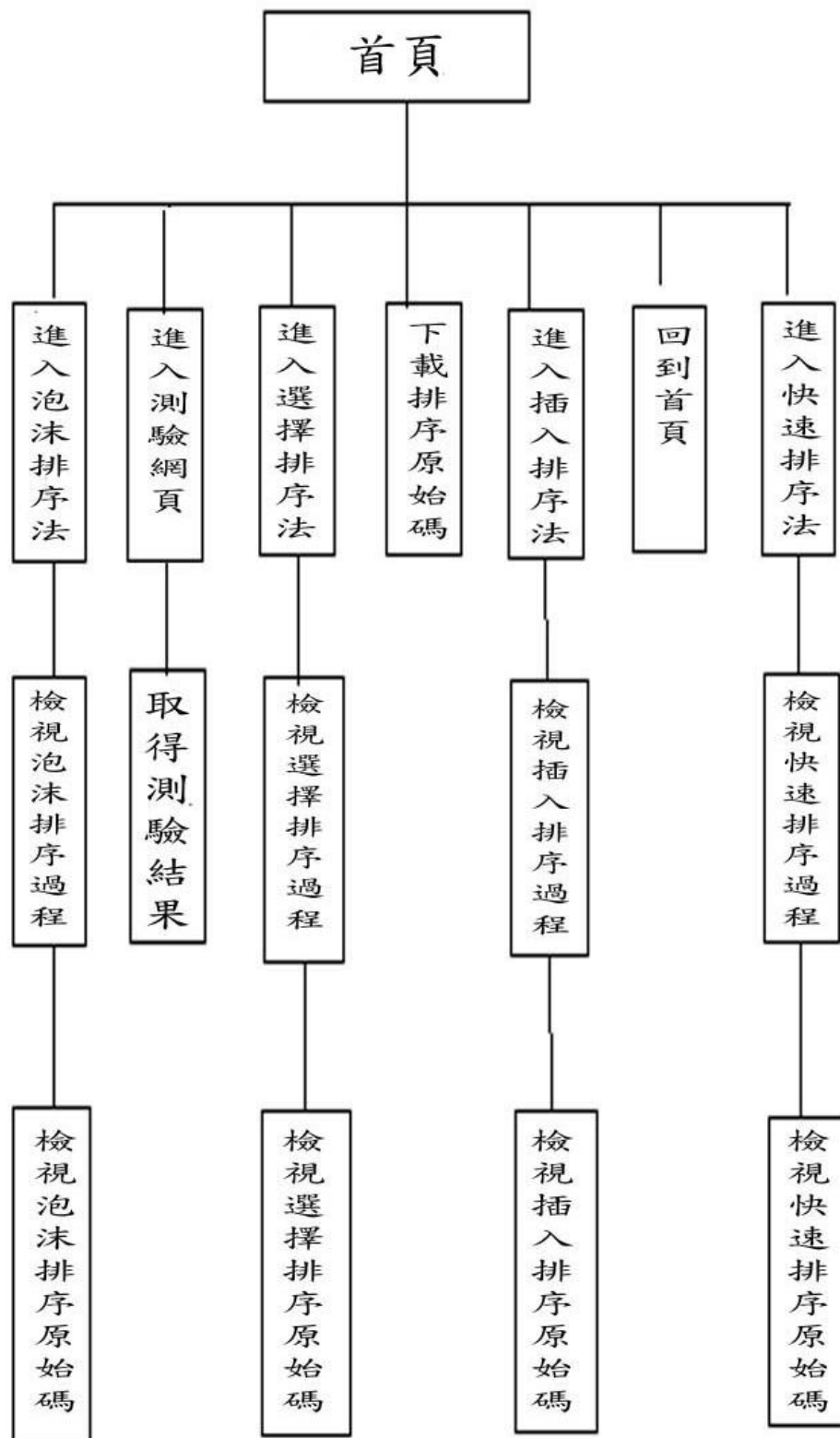


圖 3-2 網站架構圖

有關各個排序過程的圖形化實作會再接下來的第四章中作詳盡的描述，測驗網頁，是用 Perl 及 HTML 所完成的指算是一個小小的練習。

第四章 程式實作

4.1 程式概論

本程式的重點在以 Applet 實作 Sort 並以圖形表示過程，所以會將重點放在 Java 程式的寫作上，在其他地方的著墨會減少或是不談。

4.2 程式版面的配置及按鍵實作

首先來說明版面上的文字及按鍵還有文字輸入框的實作方法

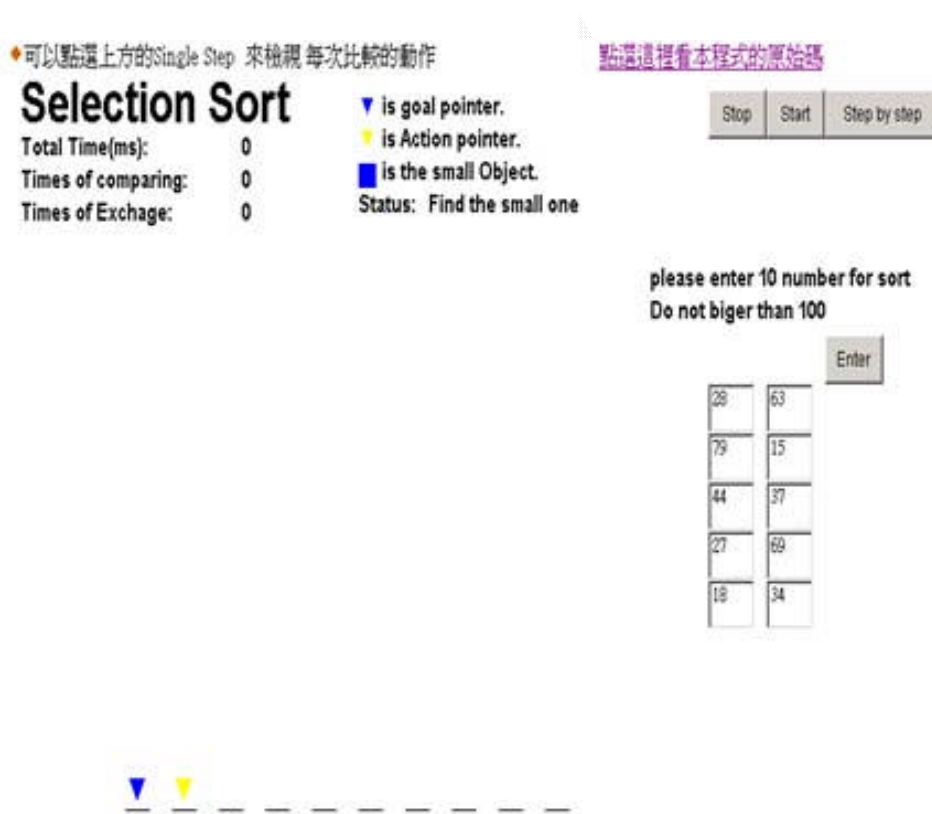


圖 4-1 已完成的版面

如圖 4-1 第一步我們將所有要顯示在版面上的文字、按鍵、文字輸入框，都先宣告型態，然後設定一個變數名稱也就是 code4-1 這段程式碼

code 4-1:

```
static final String message = "Selection Sort ";
static final String message1 = "Total Time(ms):";
static final String message2 = "Status:";
static final String message3 = "Get the small one Exange!!!";
static final String message4 = "Find the small one";
static final String message6 = "Times of comparing:";
static final String message7 = "Times of Exchange:";
static final String message8 = "Sorting Complete!!!";
static final String arrow = " ";
static final String com1 = "is goal pointer.";
static final String com2 = "is Action pointer.";
static final String com3 = "is the small Object.";
static final String message9 = "please enter 10 number for sort";
static final String message10 = "Do not bigger than 100";
//上面這個部分的是有關純文字資料的宣告
TextField text0=new TextField("28");
TextField text1=new TextField("63");
TextField text2=new TextField("79");
TextField text3=new TextField("15");
TextField text4=new TextField("44");
TextField text5=new TextField("37");
TextField text6=new TextField("27");
TextField text7=new TextField("69");
TextField text8=new TextField("18");
TextField text9=new TextField("34");
//上面這個部分是有關文字輸入框的宣告，引號內的數字為預設的資料
```

好像沒有看到按鍵的宣告？！其實按鍵的宣告也不一定要寫在這不過，我還是有寫了一個按鍵的宣告在這。

```
Button ente=new Button("Enter");
//這個是宣告按鍵的方法 引號內的文字代表按鍵上會出現的文字。
```

宣告完有關各項按鍵、文字、文字輸入框後就可以將他們排在版面上，不過有一點很重要是在配置版面的時候，要注意一點在配置的座標上是以左上角為(0,0)向右方及下方增加如圖 4-2 所示

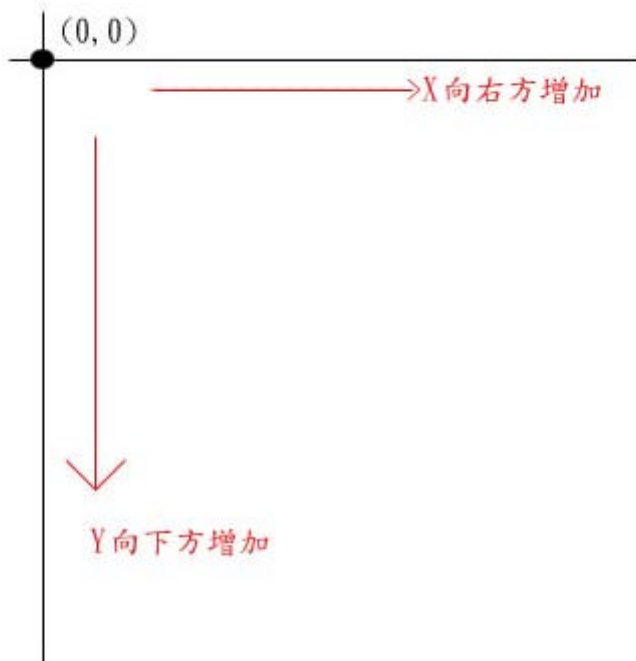


圖 4-2 座標示意圖

知道座標是怎麼增加了 就可以正式開始將該放上版面的元件放上去了放上去的方法如 code4-2 這段程式碼

code4-2:

```
fFont=new Font ("arial",Font.BOLD,15);
ffFont=new Font ("arial",Font.BOLD,20);
bffFont=new Font ("arial",Font.BOLD,35);
//這裡是描述字型的資料 Font( " 字型名稱 " ,字型樣式,大小)
stop=new Button("Stop");
stop.setBounds(600,10,50,30);
this.add(stop);
stop.addActionListener(this);
start=new Button("Start");
start.setBounds(650,10,50,30);
this.add(start);
start.addActionListener(this);
one=new Button("Step by step");
one.setBounds(700,10,100,30);
```

```
this.add(one);
one.addActionListener(this);
ente.setBounds(700,160,50,30);
this.add(ente);
ente.addActionListener(this);
//上面這段是按鍵的宣告 設置 設定當滑鼠點下時應該要工作
text0.setBounds(600,190,40,30);
this.add(text0);
text1.setBounds(650,190,40,30);
this.add(text1);
text2.setBounds(600,220,40,30);
this.add(text2);
text3.setBounds(650,220,40,30);
this.add(text3);
text4.setBounds(600,250,40,30);
this.add(text4);
text5.setBounds(650,250,40,30);
this.add(text5);
text6.setBounds(600,280,40,30);
this.add(text6);
text7.setBounds(650,280,40,30);
this.add(text7);
text8.setBounds(600,310,40,30);
this.add(text8);
text9.setBounds(650,310,40,30);
this.add(text9);
//上面這段 是將之前宣告的文字框配置在版面上
```

到這 我們已經把按鍵、文字、文字輸入框成功放上版面了接下來就要進入繪圖引擎的撰寫了

4.3 繪圖引擎的撰寫

在這個部分，會因為每個排序法的因為使用的指標有所不同，而有些微差異，不過基本的架構還是相同的，我們以選擇排序來作為範本，說明繪圖引擎的寫法。

其實在 Java 上畫圖非常容易，只要你決定好了要畫的顏色、圖形樣式、圖形的位置只要幾條指令，就可以將你想要的圖形畫在應該在

的位置，以繪製矩形為例

setColor(顏色);

fillrect(int x,int,y,int,w,int h);

x,y 為矩形左上角的座標 w 為矩形寬度 h 為矩形高度

這樣子就可以再想要的位置上畫上一個填滿顏色的矩形了。

下圖中就是將文字框中的資料換為圖形的結果

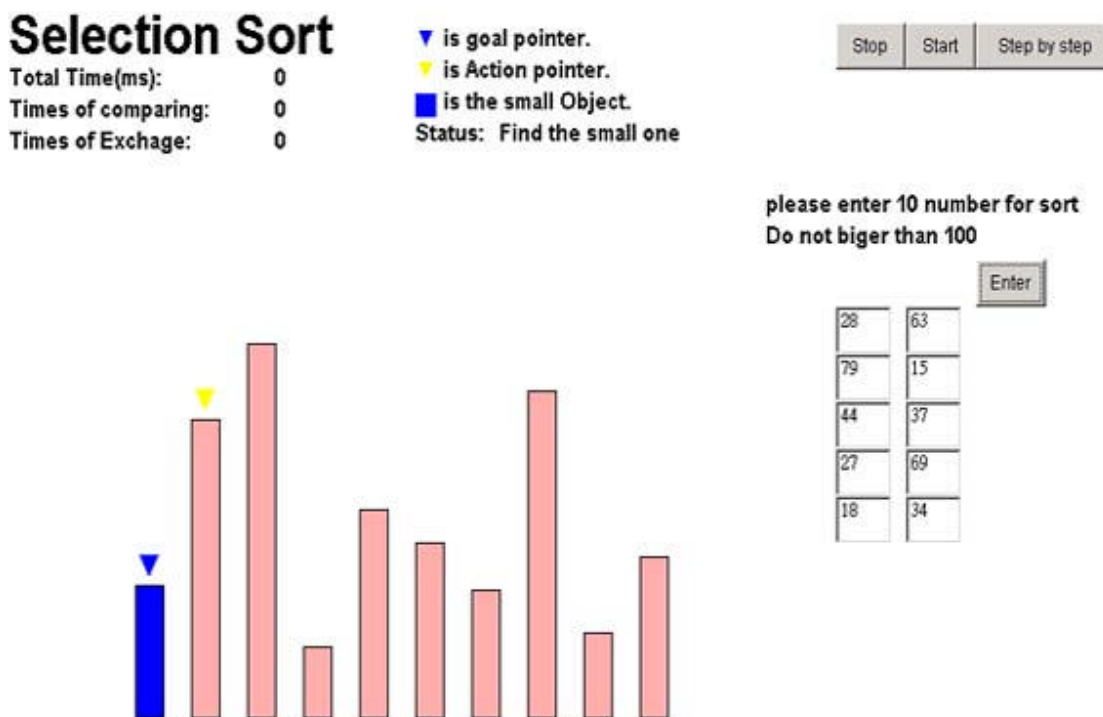


圖 4-3 將陣列資料畫上

以長條圖作為例子是因為，感覺上長條圖比較容易可以看出資料大小間的差異性 在輔以箭頭的指標及改變顏色的部分，能夠更清楚的看出資料間變動的過程。

下面是將陣列資料畫成圖形時一部份的原始碼，因為在排序過程中需要將指標凸顯出來，所以會碰到指標時的狀況，額外的標示出來而當兩個指標碰在一起的時候也會發生問題，code4-3 是當指標在同一個位置的時候我所寫的程式碼。

Code4-3:

```

if (i==h1&& i==s1)
{
    g.setColor(Color.white);
    g.fillRect(x,150,25,400);
    g.setColor(Color.blue);
    g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.yellow);
    g.setFont(fffont);
    g.drawString(arrow,x,145+ar[i]*3);
}

```

//這是當藍色的矩形和黃色箭頭碰在一起時我們必須將黃色的箭頭畫在藍色的矩形上。同樣的當藍色矩形和藍色箭頭碰在一起的時候，我們也另外畫一個矩形 並將藍色的箭頭放在矩形之上

```

else if (i==h1&& i==l1)
{
    g.setColor(Color.white);
    g.fillRect(x,150,25,400);
    g.setColor(Color.pink);
    g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.yellow);
    g.setFont(fffont);
    g.drawString(arrow,x,145+ar[i]*3);
    g.setColor(Color.blue);
    g.setFont(fffont);
    g.drawString(arrow,x,125+ar[i]*3);
}

```

//上面這段的問題就比較大了，當兩個指標碰在一起的時候 會因為你先畫哪一個而造成箭頭疊在一起 先畫的會被蓋過去所以我改將兩個箭頭以上下的方式配置 這樣看起來就不會出現箭頭失蹤的問題了。

到這裡我們已經完成了繪圖引擎了接下來就是將排序的方法裝進程式裡

4.4 排序演算法的輸入

當要開始輸入排序演算法時這個部分的困難度就下降很多了，由於已經完成了繪圖引擎，接下來就可以使用她了，要注意的是由於電腦的進步，現在的電腦，再處理排序時因為資料量不多，所以造成了幾乎是剛剛按下開始，就已經排完的結果。所以會在排序的過程中，加上幾個暫停的動作，以免眼一花什麼都沒難到的問題產生

4.4.1 Bubble Sort 演算法

我寫的 Bubble Sort 是從右方開始慢慢的將比較小的資料向陣列的左方移動當跑完第一次，可以將最小的資料留在左方，然後將左方的指標加 1 表示這筆資料已經可以不需要在加入比較的行列了，就這樣慢慢的將資料排序完成。

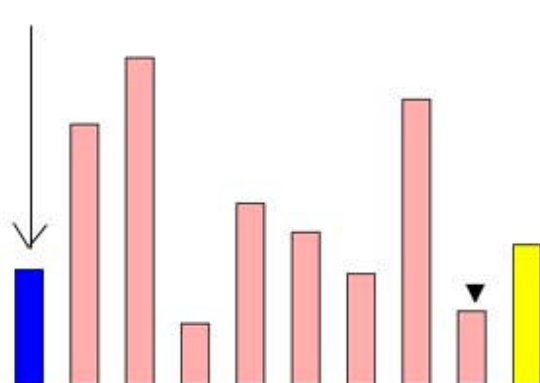
◆ 可以點選上方的Single Step 來檢視 每次比較的動作
 點選這裡看本程式的原始碼

Bubble Sort

Total Time(ms): 1200
 Times of comparing: 1
 Times of Exchange: 0

■ is goal pointer.
■ is Action pointer.
▼ is current cmparing Object.
 Status: The ▼ item is smaller then the yellow item. Non-Exange!!!

只要在這個指標的左方都算以排序好的資料



please enter 10 number for sort
Do not bigger than 100

28	63
79	15
44	37
27	69
18	34

圖 4-4 bubble sort 的指標說明

code4-4 就是這個 Bubble Sort 排序過程的部分程式碼

Code4-4:

```
for (;tmpi<a.length;tmpi++ )
{
    l1=tmpi;
    for (tmpj = a.length-1; tmpj>tmpi; tmpj--)
    {
        h1=tmpj;
        exg=false;
        //在這裡要記得把 exg 這個布林變數設為 false 不然會造成狀態欄
        錯誤
        pause2();//這個就是所加上的暫停
        if (a[tmpj] > a[tmpj-1])
        {
            exg=true;
            //這個表示有作交換動作 在狀態那欄應該要表示有交換動作
            ecount=ecount+1;
            int T = a[tmpj];
            a[tmpj] = a[tmpj-1];
            a[tmpj-1] = T;
        }
        counter=counter+1200;
        count=count+1;
        pause();//這個是所設定的第 2 個暫停
    }
}
complete=true;//這個是標示排序完成的布林變數
l1=l1-1;
repaint();
```

4.4.2 Selection Sort 演算法

Selection Sort 是從陣列的後方開始將陣列中最小的資料找出來，然後將這筆資料向前交換，將前面的指標+1 表示指標前資料已經排序完成，在接下來的排序中，這些資料都不需要再次排序，這樣就是 Selection Sort

可以點選上方的Single Step 來檢視 每次比較的動作

Selection Sort

Total Time(ms): 600
 Times of comparing: 1
 Times of Exchange: 0

▼ is goal pointer.
 ▲ is Action pointer.
 ■ is the small Object.
 Status: Find the small one

點選這裡看本程式的原始碼

Stop Start Step by step

please enter 10 number for sort
 Do not bigger than 100

Enter

28	63
79	15
44	37
27	69
18	34

這個箭頭的左邊表示已經排好的資料

<< 這個顏色的矩陣表示目前找到的最小資料

圖 4-5 selection sort 的指標說明

Code4-5 這個就是 Selection Sort 排序過程的部分程式碼

Code4-5:

```

for ( ;tmpi<a.length;tmpi++ )
{
    l1=tmpi;
    s1=a.length-1;//一開始先將最小的位置預設為陣列的最右方
    for (tmpj = a.length-1; tmpj>=tmpi; tmpj--)
    {
        exg=false;
        h1=tmpj;
        if (a[tmpj] > a[s1])
        {
            s1=tmpj;//將目前找到最小的資料用指標 標示出來
        }
        count=count+1;
        pause();
    }
}
    
```

```

exg=true;
pause();
int t=a[s1];
a[s1]=a[tmpi];
a[tmpi]=t;
//此時已經可以確定 s1 所指的位置為最小的資料
此時將最小的資料，和未排序資料的第一項作交換
}
complete=true;
l1=l1-1;
repaint();
}
    
```

4.4.3 Insertion Sort 演算法

我們從陣列的第一筆和第 2 筆資料開始先排好這兩筆資料接著比較下一筆資料，如果下一筆資料，比現在已經排序過的最後一筆資料還小，則開始自己排序資料的後方一項一項比對，找到應該在的位置，將資料插入。

可以點選上方的Single Step 來檢視 每次比較的動作 [點選這裡看本程式的原始碼](#)

Insertion Sort

Total Time(ms): 600
 Times of comparing: 1
 Times of Exchange: 0

■ is goal pointer.
■ is Action pointer.
 ▼ is current cmparing Object.
 Status: The blue item is bigger then the ▼ item. Non-Exange!!!

在這筆資料之前的資料以排序完成

please enter 10 number for sort
Do not bigger than 100

28	63
79	15
44	37
27	69
18	34

圖 4-6 Insertion Sort 的指標說明

Code4-6 是這個 Insertion Sort 排序過程的部分程式碼

Code4-6:

```
for (;tmpi<a.length;tmpi++)
{
l1=tmpi;
//將指標定在目前已排序的下一筆資料上
tmpda=a[tmpi];
exg = false;
for (tmpj=tmpi-1;tmpj>=0&&tmpda > a[tmpj]; tmpj--)
//若發現已排序的最後一筆資料筆下一筆大則開始找應該要插入的
位置
{
h1 = tmpj;
pause2();
exg = true;
count = count + 1;
ecount = ecount + 1;
a[tmpj + 1] = a[tmpj];
}
a[tmpj + 1] = tmpda;
//將過程中被替換掉的資料放回應該在的位置
h1=l1-1;
pause();
}
complete=true;
repaint();
```

4.4.4 Quick Sort 演算法

將第一筆資料當成基礎將下一筆資料及最後一筆資料當成指標先從第一個指標開始向後面尋找比第一筆資料大的資料出來，再從第二個指標向前找出比第一筆資料小的資料，這個時候檢查兩個指標是不是已經越過對方了，如果沒有就把兩者對調，如果有就把第一筆資料和第二個指標資料交換，這樣就可以將資料分成兩段在對前半段及後半段，在做相同的動作，循環下去就可以把資料排序完成。

●可以點選上方的Single Step 來檢視 每次比較的動作 [點選這裡看本程式的原始碼](#)

Quick Sort

Total Time(ms): 600
 Times of comparing: 1
 Times of Exchange: 0

▼ is the pointer to find big one
 ▼ is the pointer to find small one
 ■ is the compare one
 Status: To find the item which is smaller then blue item

Stop Start Step by step

用來找出比藍色矩陣資料大的指標

用來找出比藍色矩陣資料小的指標

基準值 >>

please enter 10 number for sort
 Do not bigger than 100

Enter

28	63
79	15
44	37
27	69
18	34

圖 4-7 Quick Sort 的指標說明

Code4-7 是 Quick Sort 排序過程的部分程式碼

Code4-7:

```
int fir_tmp, fin_tmp, tmp, flag;
m1 = fir;//將第一筆資料設為基準
if (fir < fin)
{
fir_tmp = fir + 1;
l1 = fir_tmp;
fin_tmp = fin;
h1 = fin_tmp;
tmp = a[fir];
flag = 1;
while (flag == 1)
{
exg=false;
comps=false;
compb=false;
while (a[fir_tmp] > tmp)//找出比第一筆資料大的資料
{
```

```

compb=true;
pause2();
fir_tmp = fir_tmp + 1;
l1 = fir_tmp;
}
while (a[fin_tmp] < tmp)//找出比第一筆資料小的資料
{
comps=true;
pause2();
fin_tmp = fin_tmp - 1;
h1 = fin_tmp;
}
if (fir_tmp < fin_tmp)//檢查兩個指標有沒有越過對方了
{
pause2();
tmpi = a[fir_tmp];
a[fir_tmp] = a[fin_tmp];
a[fin_tmp] = tmpi;//沒有 交換兩個指標的資料
}
else {
flag = 0;//有 停止迴圈
}
pause();
}
pause2();
tmpj = a[fir];
a[fir] = a[fin_tmp];
a[fin_tmp] = tmpj;//將資料分為前半段 和 後半段
sort(a, fir, fin_tmp - 1);//將前半段資料重做排序
sort(a, fin_tmp + 1, fin);//將後半段資料重做排序
}

```

4.5 測驗題實作

perl 是個很方便的語言我們只要在開頭的地方，加上
print "content-type:text/html\n\n"; 這一行指令，就可以利用
print “html 語法”；這樣的方法來使用 html 語法完成表單的設
置，舉個這次實際使用上的例子由於這次再做測驗題時，選項的決定
我用的是 radio button 所以在每個選項的前面加上
print "<input type=\"hidden\" name=\"quest0\" value=\"1\">";

這樣的一行敘述，就可以在選項前面加上一個可以點選的框，並且由於這個 radio button 的特性同一個名字的只會有一個選項被圈選，所以每次送出時一個名字只會有一個值，也就是單選題。

另外 perl 可以直接在程式中作數值的計算，亂數的取得，條件式的控制，只要和 c 差不多的寫法，可以做出一樣的效果，像這次用到的隨機取題目 就是用一條 `$k=int(rand(15));` 就可以取得 0-15 間的數字。這樣就可以幫助完成隨機取得題目的完成。

第五章 發生的問題及心得

在程式的寫作過程，一定會發生幾項問題，在這裡把她說明出來，並將解決的方法一並提出

5.1 發生的問題

Q：整個程式在一開始就發生了一個問題 就是座標問題，一開始程式在實作的時候，我發現要是直接把陣列的資料畫圖，所得到的結果會是一個向下長的長條圖，這樣看起來就不像正常的長條圖了。

A：也許已經有人發現的解決方法了，其實我在排序的過程中真正在排序的是圖形上方的白色部分，也就是說我要找看起來最小的資料，其實真正要找的是最大的資料也因為如此，在手動輸入資料時，會先用 100 減去輸入的資料在將資料放進陣列排序這樣可以達到看起來正常的圖形。

Q：一開始實作的過程中發現，中文字及作為指標的箭頭經過 jacac 的編譯後會變成亂碼，試過若是以英文代替中文沒有問題，但箭頭的部分還是無解。

A：問過幾個有寫 Java 的朋友，他們給我的答案都是將 J2SDK 更換到 1.4 以上版本，再重新編譯一次，箭頭的問題就可以解決了，在我的嘗試之下，果然成功的把箭頭表示出來，但中文字的部分還是無解，所以才會用英文標示狀態及各項資料。

5.2 心得

我覺得專題的寫作目的，應該是要檢視我們在大學的四年內學到了什麼東西，並且藉著這次的機會，將妳有興趣的部分在做更加一步精進，這次我選擇了用 Java 來完成這次的專題，雖然是以前沒有接觸過太多次的東西，但在這次完成專題的過程中，可以說是獲益良多。雖然在 Java 程式的寫作上，還有很多東西需要再學習，但這也算是已經跨出了第一步，接下來的就是藉著這次的經驗，自我再多做練習。在這麼多優良的專題下，這份專題並不是很完美的，但我覺得，我所得知識和經驗，是不能衡量的，在寫 java code 的過程中，找回了以前發現問題會自己去找出答案來的心情，當我有問題時，也確切的感受到了朋友的關懷，所有的 code 都是一個字一個字慢慢敲出來的。

另外免不了的要感謝一下指導教授黃溪春老師，因為我的起步算晚的，但老師能在百忙中，還能抽出時間幫我看我的成品，很多時候都覺得很不好意思。

參考書目

- [1]柯溫釗編著,Java 學習經典,第三波,2000/12
- [2]R.Allen Wyke&Donald B.Thomas 著,吳東賢譯,Perl 程式設計入門手冊,美商麥格羅.希爾國際股份有限公司台灣分公司,2001/6
- [3]黃國欽,最新 Java 程式設計入門與實作,松崗圖書,2002/5
- [4]蔡明志,資料結構 使用 C 語言,碁峰,2002/6

附錄 A Bubble Sort 的 Applet 程式實作

Source Code

```
import java.awt.*;  
import java.awt.event.*;  
import java.io.InputStream;  
import java.util.Hashtable;  
import java.net.*;
```

```
import java.applet.*;

public class bubblesort extends java.applet.Applet implements
Runnable,MouseListener,ActionListener
{
    static final String message = "Bubble Sort ";
    static final String message1 = "Total Time(ms):";
    static final String message2 = "Status:";
    static final String message3 = "The    item is bigger then the yellow
item. Exchange!!!";
    static final String message4 = "The    item is smaller then the yellow
item. Non-Exchange!!!";
    static final String message6 = "Times of comparing:";
    static final String message7 = "Times of Exchange:";
    static final String message8 = "Sorting Complete!!!";
    static final String arrow = " ";
    static final String com1 = "is goal pointer.";
    static final String com2 = "is Action pointer.";
    static final String com3 = "is current comparing Object.";
    static final String message9 = "please enter 10 number for sort";
    static final String message10 = "Do not bigger than 100";
    TextField text0=new TextField("28");
    TextField text1=new TextField("63");
    TextField text2=new TextField("79");
    TextField text3=new TextField("15");
    TextField text4=new TextField("44");
    TextField text5=new TextField("37");
    TextField text6=new TextField("27");
    TextField text7=new TextField("69");
    TextField text8=new TextField("18");
    TextField text9=new TextField("34");

    Button ente=new Button("Enter");
    String ti;
    String si;
    String ii;
    String s;
    private Font ffont;
    private Font fffont;
    private Font bffont;
    int h1=1;
```

```
int l1=0;
int x=0;
    int k=0;
    int seth=0;
int setl=0;
int tmpi=0;
int tmpj=0;
int counter=0;
int count=0;
int[] ar=new int[ ]{100,100,100,100,100,100,100,100,100,100};
int ecount=0;
boolean exg=false;
boolean success=false;
boolean step=false;
boolean stopacc=false;
boolean run=true;
boolean warn=false;
boolean initial=true;
boolean complete=false;
boolean comp=true;
Button stop;
Button start;
Button one;

public void init()
{

    ffont=new Font ("arial",Font.BOLD,15);
    fffont=new Font ("arial",Font.BOLD,20);
        bffont=new Font ("arial",Font.BOLD,35);
setBackground(Color.white);
addMouseListener(this);
this.setLayout(null);
stop=new Button("Stop");
stop.setBounds(600,10,50,30);
this.add(stop);
stop.addActionListener(this);
start=new Button("Start");
start.setBounds(650,10,50,30);
this.add(start);
```

```
start.addActionListener(this);
one=new Button("Step by step");
one.setBounds(700,10,100,30);
this.add(one);
one.addActionListener(this);
    ente.setBounds(700,160,50,30);
    this.add(ente);
    ente.addActionListener(this);
    text0.setBounds(600,190,40,30);
    this.add(text0);
    text1.setBounds(650,190,40,30);
    this.add(text1);
    text2.setBounds(600,220,40,30);
    this.add(text2);
    text3.setBounds(650,220,40,30);
    this.add(text3);
    text4.setBounds(600,250,40,30);
    this.add(text4);
    text5.setBounds(650,250,40,30);
    this.add(text5);
    text6.setBounds(600,280,40,30);
    this.add(text6);
    text7.setBounds(650,280,40,30);
    this.add(text7);
    text8.setBounds(600,310,40,30);
    this.add(text8);
    text9.setBounds(650,310,40,30);
    this.add(text9);

}
Thread algsort=null;

public void start()
{
}
public void stop()
{
    algsort=null;
}
public void run()
{
```

```
        run=true;
        initial=false;
        sort(ar);
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}

    public void paint(Graphics g)
    {
        int x=100;
        int y=100;
        g.setColor(Color.black);
        g.setFont(bfont);
        g.drawString(message,10,30);
        g.setFont(ffont);
        g.drawString(message1,10,50);
        g.setFont(ffont);
        g.drawString(message2,300,85);
        g.setFont(ffont);
        g.drawString(message6,10,70);
        g.setFont(ffont);
        g.drawString(message7,10,90);
            g.setFont(ffont);
        g.drawString(message9,550,130);
            g.setFont(ffont);
        g.drawString(message10,550,150);

        g.setColor(Color.blue);
        g.fillRect(300,15,15,15);
        g.setFont(ffont);
        g.setColor(Color.black);
        g.drawString(com1,320,25);

        g.setColor(Color.yellow);
        g.fillRect(300,35,15,15);
        g.setFont(ffont);
        g.setColor(Color.black);
```

```
g.drawString(com2,320,45);

g.setFont(null);
g.setColor(Color.black);
g.drawString(arrow,300,65);
g.drawString(com3,320,65);

int tmp=0;
if(complete==true)
{
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message8,360,85);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString("Result:",500,370);
    for (int i=0;i<ar.length;i++)
    {
        k=100-ar[i];
        s = String.valueOf(k);
        g.drawString(s,500+30*i,390);
    }

    }else if(exg==true)
{
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message3,360,85);

}else if(exg==false){
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message4,360,85);
```

```

    }
        ti=String.valueOf(counter);
        g.setColor(Color.white);
        g.fillRect(200,30,100,20);
        g.setColor(Color.black);
        g.setFont(ffont);
        g.drawString(ti,200,50);
        si=String.valueOf(count);
        g.setColor(Color.white);
        g.fillRect(200,50,100,20);
        g.setColor(Color.black);
        g.setFont(ffont);
        g.drawString(si,200,70);
        ii=String.valueOf(ecount);
        g.setColor(Color.white);
        g.fillRect(200,70,100,20);
        g.setColor(Color.black);
        g.setFont(ffont);
        g.drawString(ii,200,90);

for(int i=0;i<ar.length;i++)
{
    g.setColor(Color.white);
    g.fillRect(x,130,90,20);
    if(h1==i&&exg==true)
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.fillRect(x-40,150,25,400);
        if(i-1==l1)
        {
            g.setColor(Color.yellow);
            g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
            g.setColor(Color.black);
            g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);

            g.setColor(Color.blue);
            g.fillRect(x-40,150+ar[i-1]*3,20,300-ar[i-1]*3);
            g.setColor(Color.black);
            g.drawRect(x-40,150+ar[i-1]*3,20,300-ar[i-1]*3);
        }
    }
}

```

```

        g.setFont(fffont);
        g.drawString(arrow,x-37,145+ar[i-1]*3);
    }else{

        g.setColor(Color.yellow);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);

        g.setColor(Color.pink);
        g.fillRect(x-40,150+ar[i-1]*3,20,300-ar[i-1]*3);
        g.setColor(Color.black);
        g.drawRect(x-40,150+ar[i-1]*3,20,300-ar[i-1]*3);

        g.setFont(fffont);
        g.drawString(arrow,x-37,145+ar[i-1]*3);
    }
}
if(h1==i&&exg==false)
{
    g.setColor(Color.white);
    g.fillRect(x,150,25,ar[i]*3);

    g.setColor(Color.yellow);
    g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setFont(fffont);
    g.drawString(arrow,x-37,145+ar[i-1]*3);
}
if(l1==i)
{
    if(i!=l1+1)
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,ar[i]*3);
    }
}

```



```
        g.setColor(Color.blue);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    }
    if(i!=h1&&i!=l1)
    {
        if(i!=h1-1)
        {
            g.setColor(Color.white);
            g.fillRect(x,150,25,ar[i]*3);
        }

        g.setColor(Color.pink);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    }

    x=x+40;
}

}
public void update(Graphics g)
{
    paint(g);
}

void pause()
{

    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(600);} catch (InterruptedException e){}
}
void pause2()
```

```
{

    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(800);} catch (InterruptedException e){}
}

public void sort(int a[])
{
    for (;tmpi<a.length;tmpi++ )
    {
        l1=tmpi;

        for (tmpj = a.length-1; tmpj>tmpi; tmpj--)
        {

            h1=tmpj;
            exg=false;
            pause2();
            if (a[tmpj] > a[tmpj-1])
            {
                exg=true;
                ecount=ecount+1;
                int T = a[tmpj];
                a[tmpj] = a[tmpj-1];
                a[tmpj-1] = T;

            }
            counter=counter+1200;
            count=count+1;
            pause();

            if(step==true)
            {

                algsort.suspend();
```

```
        }  
    }  
  
    }  
    complete=true;  
    l1=l1-1;  
    repaint();  
}  
  
public void actionPerformed(ActionEvent e) {  
    Object src = e.getSource();  
    if( src==stop ) {  
  
        run=false;  
        algsort.suspend();  
        return;  
    }else if(src==start)  
    {  
        run=true;  
        step=false;  
        if (algsort == null)  
        {  
algsort = new Thread(this);  
algsort.start();  
        }else{  
algsort.resume();  
        }  
  
        return;  
    }else if (src==one)  
    {  
  
        step=true;  
  
        if (algsort == null)  
        {  
algsort = new Thread(this);  
algsort.start();  
        }else  
    }
```

```

        {
            algsort.resume();
        }
        return;
    }else if (src==ente)
    {
        ar[0]=100- Integer.parseInt(text0.getText());
        ar[1]=100- Integer.parseInt(text1.getText());
        ar[2]=100- Integer.parseInt(text2.getText());
        ar[3]=100- Integer.parseInt(text3.getText());
        ar[4]=100- Integer.parseInt(text4.getText());
        ar[5]=100- Integer.parseInt(text5.getText());
        ar[6]=100- Integer.parseInt(text6.getText());
        ar[7]=100- Integer.parseInt(text7.getText());
        ar[8]=100- Integer.parseInt(text8.getText());
        ar[9]=100- Integer.parseInt(text9.getText());
        repaint();
    }
}
}
}

```

附錄 B Selection Sort 的 Applet 程式實作

Source Code

```

import java.awt.*;
import java.awt.event.*;
import java.io.InputStream;
import java.util.Hashtable;
import java.net.*;
import java.applet.*;

public class sesort extends java.applet.Applet implements
Runnable,MouseListener,ActionListener
{
    static final String message = "Selection Sort ";
    static final String message1 = "Total Time(ms):";
    static final String message2 = "Status:";
    static final String message3 = "Get the small one Exange!!!";
    static final String message4 = "Find the small one";

```

```

static final String message6 = "Times of comparing:";
static final String message7 = "Times of Exchange:";
static final String message8 = "Sorting Complete!!!";
static final String arrow = " ";
static final String com1 = "is goal pointer.";
static final String com2 = "is Action pointer.";
static final String com3 = "is the small Object.";
    static final String message9 = "please enter 10 number for sort";
    static final String message10 = "Do not bigger than 100";
    TextField text0=new TextField("28");
    TextField text1=new TextField("63");
    TextField text2=new TextField("79");
    TextField text3=new TextField("15");
    TextField text4=new TextField("44");
    TextField text5=new TextField("37");
    TextField text6=new TextField("27");
    TextField text7=new TextField("69");
    TextField text8=new TextField("18");
    TextField text9=new TextField("34");

    Button ente=new Button("Enter");
String ti;
String si;
String ii;
    String s;
private Font ffont;
private Font fffont;
    private Font bffont;
int h1=1;
int l1=0;
    int s1=0;
int x=0;
    int k=0;
int seth=0;
int setl=0;
int tmpi=0;
int tmpj=0;
int counter=0;
int count=0;
int[] ar=new int[ ]{100,100,100,100,100,100,100,100,100,100};
int ecount=0;

```

```
boolean exg=false;
boolean success=false;
boolean step=false;
boolean stopacc=false;
boolean run=true;
boolean warn=false;
boolean initial=true;
boolean complete=false;
boolean comp=true;
Button stop;
Button start;
Button one;
```

```
public void init()
{
    ffont=new Font ("arial",Font.BOLD,15);
    fffont=new Font ("arial",Font.BOLD,20);
        bffont=new Font ("arial",Font.BOLD,35);
    setBackground(Color.white);
    addMouseListener(this);
    this.setLayout(null);
    stop=new Button("Stop");
    stop.setBounds(600,10,50,30);
    this.add(stop);
    stop.addActionListener(this);
    start=new Button("Start");
    start.setBounds(650,10,50,30);
    this.add(start);
    start.addActionListener(this);
    one=new Button("Step by step");
    one.setBounds(700,10,100,30);
    this.add(one);
    one.addActionListener(this);
        ente.setBounds(700,160,50,30);
        this.add(ente);
        ente.addActionListener(this);
    text0.setBounds(600,190,40,30);
    this.add(text0);
    text1.setBounds(650,190,40,30);
```

```

        this.add(text1);
        text2.setBounds(600,220,40,30);
        this.add(text2);
        text3.setBounds(650,220,40,30);
        this.add(text3);
        text4.setBounds(600,250,40,30);
        this.add(text4);
        text5.setBounds(650,250,40,30);
        this.add(text5);
        text6.setBounds(600,280,40,30);
        this.add(text6);
        text7.setBounds(650,280,40,30);
        this.add(text7);
        text8.setBounds(600,310,40,30);
        this.add(text8);
        text9.setBounds(650,310,40,30);
        this.add(text9);
    }
    Thread algsort=null;

    public void start()
    {
    }
    public void stop()
    {
        algsort=null;
    }
    public void run()
    {
        run=true;
        initial=false;
        sort(ar);
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}

```



```

public void paint(Graphics g)
{
    int x=100;
    int y=100;
    g.setColor(Color.black);
    g.setFont(bfont);
    g.drawString(message,10,30);
    g.setFont(ffont);
    g.drawString(message1,10,50);
    g.setFont(ffont);
    g.drawString(message2,300,85);
    g.setFont(ffont);
    g.drawString(message6,10,70);
    g.setFont(ffont);
    g.drawString(message7,10,90);
        g.setFont(ffont);
    g.drawString(message9,550,130);
        g.setFont(ffont);
    g.drawString(message10,550,150);

    g.setFont(null);
        g.setColor(Color.blue);
        g.drawString(arrow,300,25);
    g.setFont(ffont);
    g.setColor(Color.black);
    g.drawString(com1,320,25);

    g.setFont(null);
        g.setColor(Color.yellow);
        g.drawString(arrow,300,45);
    g.setFont(ffont);
    g.setColor(Color.black);
    g.drawString(com2,320,45);

        g.setColor(Color.blue);
    g.fillRect(300,55,15,15);
    g.setColor(Color.black);
    g.drawString(com3,320,65);

    int tmp=0;

```



```

if(complete==true)
{
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message8,360,85);
        g.setColor(Color.black);
        g.setFont(ffont);
        g.drawString("Result:",500,370);
        for (int i=0;i<ar.length;i++)
        {
            k=100-ar[i];
            s = String.valueOf(k);
            g.drawString(s,500+30*i,390);
        }

    }else if(exg==true)
{
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message3,360,85);

}else if(exg==false){
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message4,360,85);

}

    ti=String.valueOf(counter);
    g.setColor(Color.white);
    g.fillRect(200,30,100,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(ti,200,50);
    si=String.valueOf(count);
    g.setColor(Color.white);

```

```

g.fillRect(200,50,100,20);
g.setColor(Color.black);
g.setFont(ffont);
g.drawString(si,200,70);
ii=String.valueOf(ecount);
g.setColor(Color.white);
g.fillRect(200,70,100,20);
g.setColor(Color.black);
g.setFont(ffont);
g.drawString(ii,200,90);

```

```

for(int i=0;i<ar.length;i++)
{
    g.setColor(Color.white);
    g.fillRect(x,130,90,20);
    if (i==h1&&i==s1)
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.blue);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.yellow);
        g.setFont(ffont);
        g.drawString(arrow,x,145+ar[i]*3);
    }
    else if(i==l1&&i==s1)
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.blue);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.blue);
        g.setFont(ffont);
        g.drawString(arrow,x,145+ar[i]*3);
    }
    else if (i==h1&&i==l1)

```

```

    {
        g.setColor(Color.white);
        g.fillRect(x, 150, 25, 400);
        g.setColor(Color.pink);
        g.fillRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
        g.setColor(Color.yellow);
        g.setFont(fffont);
        g.drawString(arrow, x, 145+ar[i]*3);
        g.setColor(Color.blue);
        g.setFont(fffont);
        g.drawString(arrow, x, 125+ar[i]*3);
    }
else if (i==h1)
{
    g.setColor(Color.white);
    g.fillRect(x, 150, 25, 400);
    g.setColor(Color.pink);
    g.fillRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
    g.setColor(Color.yellow);
    g.setFont(fffont);
    g.drawString(arrow, x, 145+ar[i]*3);

}
else if(i==l1)
{
    g.setColor(Color.white);
    g.fillRect(x, 150, 25, 400);
    g.setColor(Color.pink);
    g.fillRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
    g.setColor(Color.blue);
    g.setFont(fffont);
    g.drawString(arrow, x, 145+ar[i]*3);

}
else if (i==s1)
{

```

```

        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.blue);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    }
    else
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.pink);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    }

    x=x+40;
}

}

public void update(Graphics g)
{
    paint(g);
}

void pause()
{

    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(600);} catch (InterruptedException e){}
}

void pause2()
{

    if (algsort != null)
    {

```

```

        repaint();
    }
    try {Thread.sleep(800);} catch (InterruptedException e){}
}

public void sort(int a[])
{
    for (;tmpi<a.length;tmpi++ )
    {
        l1=tmpi;
        s1=a.length-1;
        for (tmpj = a.length-1; tmpj>=tmpi; tmpj--)
        {
            exg=false;
            h1=tmpj;
            if (a[tmpj] > a[s1])
            {
                s1=tmpj;
            }
            count=count+1;
            pause();
            counter=counter+600;
            if(step==true)
            {
                algsort.suspend();
            }
        }
        exg=true;
        pause();
        counter=counter+600;
        ecoun=ecoun+1;
        int t=a[s1];
        a[s1]=a[tmpj];
        a[tmpj]=t;
    }
    complete=true;
    l1=l1-1;
    repaint();
}

```

```

public void actionPerformed(ActionEvent e) {
Object src = e.getSource();
    if( src==stop ) {

run=false;
algsort.suspend();
return;
}else if(src==start)
{
    run=true;
    step=false;
    if (algsort == null)
        {
algsort = new Thread(this);
algsort.start();
}else{
algsort.resume();
    }

    return;
}else if (src==one)
{

    step=true;

    if (algsort == null)
        {
algsort = new Thread(this);
algsort.start();
}else
{
    algsort.resume();
}
return;
}
}else if (src==ente)
{
    ar[0]=100- Integer.parseInt(text0.getText());
    ar[1]=100- Integer.parseInt(text1.getText());
    ar[2]=100- Integer.parseInt(text2.getText());
    ar[3]=100- Integer.parseInt(text3.getText());
    ar[4]=100- Integer.parseInt(text4.getText());
}
}

```

```

        ar[5]=100- Integer.parseInt(text5.getText());
        ar[6]=100- Integer.parseInt(text6.getText());
        ar[7]=100- Integer.parseInt(text7.getText());
        ar[8]=100- Integer.parseInt(text8.getText());
        ar[9]=100- Integer.parseInt(text9.getText());
        repaint();
    }

}

}

```

附錄 C Insertion Sort 的 Applet 程式實作

Source Code

```

import java.awt.*;
import java.awt.event.*;
import java.io.InputStream;
import java.util.Hashtable;
import java.net.*;
import java.applet.*;

public class insort extends java.applet.Applet implements
Runnable,MouseListener,ActionListener
{
    static final String message = "Insertion Sort ";
    static final String message1 = "Total Time(ms):";
    static final String message2 = "Status:";
    static final String message3 = "The blue item is smailler then the
item. Exange!!!";
    static final String message4 = "The blue item is biger then the
item. Non-Exange!!!";
    static final String message6 = "Times of comparing:";

```

```

static final String message7 = "Times of Exchange:";
static final String message8 = "Sorting Complete!!!";
static final String arrow = " ";
static final String com1 = "is goal pointer.";
static final String com2 = "is Action pointer.";
static final String com3 = "is current cmparing Object.";
    static final String message9 = "please enter 10 number for sort";
    static final String message10 = "Do not bigger than 100";
    TextField text0=new TextField("28");
    TextField text1=new TextField("63");
    TextField text2=new TextField("79");
    TextField text3=new TextField("15");
    TextField text4=new TextField("44");
    TextField text5=new TextField("37");
    TextField text6=new TextField("27");
    TextField text7=new TextField("69");
    TextField text8=new TextField("18");
    TextField text9=new TextField("34");

    Button ente=new Button("Enter");
String ti;
String si;
String ii;
    String s;
private Font ffont;
private Font fffont;
    private Font bffont;
int h1=1;
int l1=0;
int x=0;
    int k=0;
int seth=0;
int setl=0;
int tmpi=1;
int tmpj=0;
    int tmpda=0;
    int tmpd=0;
int counter=0;
int count=0;
int[] ar=new int[ ]{100,100,100,100,100,100,100,100,100,100};
int ecount=0;

```



```
boolean exg=false;
boolean success=false;
boolean step=false;
boolean stopacc=false;
boolean run=true;
boolean warn=false;
boolean initial=true;
boolean complete=false;
boolean comp=true;
Button stop;
Button start;
Button one;
```

```
public void init()
{
    ffont=new Font ("arial",Font.BOLD,15);
    fffont=new Font ("arial",Font.BOLD,20);
        bffont=new Font ("arial",Font.BOLD,35);
    setBackground(Color.white);
    addMouseListener(this);
    this.setLayout(null);
    stop=new Button("Stop");
    stop.setBounds(600,10,50,30);
    this.add(stop);
    stop.addActionListener(this);
    start=new Button("Start");
    start.setBounds(650,10,50,30);
    this.add(start);
    start.addActionListener(this);
    one=new Button("Step by step");
    one.setBounds(700,10,100,30);
    this.add(one);
    one.addActionListener(this);
        ente.setBounds(700,160,50,30);
        this.add(ente);
        ente.addActionListener(this);
        text0.setBounds(600,190,40,30);
        this.add(text0);
        text1.setBounds(650,190,40,30);
```

```

        this.add(text1);
        text2.setBounds(600,220,40,30);
        this.add(text2);
        text3.setBounds(650,220,40,30);
        this.add(text3);
        text4.setBounds(600,250,40,30);
        this.add(text4);
        text5.setBounds(650,250,40,30);
        this.add(text5);
        text6.setBounds(600,280,40,30);
        this.add(text6);
        text7.setBounds(650,280,40,30);
        this.add(text7);
        text8.setBounds(600,310,40,30);
        this.add(text8);
        text9.setBounds(650,310,40,30);
        this.add(text9);
    }
    Thread algsort=null;

    public void start()
    {
    }
    public void stop()
    {
        algsort=null;
    }
    public void run()
    {
        run=true;
        initial=false;
        sort(ar);
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}

```

```

public void paint(Graphics g)
{
    int x=100;
    int y=100;
    g.setColor(Color.black);
    g.setFont(bfont);
    g.drawString(message,10,30);
    g.setFont(ffont);
    g.drawString(message1,10,50);
    g.setFont(ffont);
    g.drawString(message2,300,85);
    g.setFont(ffont);
    g.drawString(message6,10,70);
    g.setFont(ffont);
    g.drawString(message7,10,90);
        g.setFont(ffont);
    g.drawString(message9,550,130);
        g.setFont(ffont);
    g.drawString(message10,550,150);

    g.setColor(Color.blue);
    g.fillRect(300,15,15,15);
    g.setFont(ffont);
    g.setColor(Color.black);
    g.drawString(com1,320,25);

    g.setColor(Color.yellow);
    g.fillRect(300,35,15,15);
    g.setFont(ffont);
    g.setColor(Color.black);
    g.drawString(com2,320,45);

    g.setFont(null);
    g.setColor(Color.black);
    g.drawString(arrow,300,65);
    g.drawString(com3,320,65);

    int tmp=0;
    if(complete==true)
    {

```

```

g.setColor(Color.white);
g.fillRect(360,70,500,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(message8,360,85);
    g.setColor(Color.black);
    g.setFont(f font);
    g.drawString("Result:",500,370);
    for (int i=0;i<ar.length;i++)
    {
        k=100-ar[i];
        s = String.valueOf(k);
        g.drawString(s,500+30*i,390);
    }

    }else if(exg==true)
{
g.setColor(Color.white);
g.fillRect(360,70,500,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(message3,360,85);

}else if(exg==false){
g.setColor(Color.white);
g.fillRect(360,70,500,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(message4,360,85);

}

    ti=String.valueOf(counter);
g.setColor(Color.white);
g.fillRect(200,30,100,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(ti,200,50);
si=String.valueOf(count);
g.setColor(Color.white);
g.fillRect(200,50,100,20);
g.setColor(Color.black);

```

```

g.setFont(ffont);
g.drawString(si,200,70);
ii=String.valueOf(ecount);
g.setColor(Color.white);
g.fillRect(200,70,100,20);
g.setColor(Color.black);
g.setFont(ffont);
g.drawString(ii,200,90);

```

```

for(int i=0;i<ar.length;i++)
{
    g.setColor(Color.white);
    g.fillRect(x,130,90,20);
    if (i==h1)
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.yellow);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setFont(ffont);
        g.drawString(arrow,x,145+ar[i]*3);
    }else if(i==l1)
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.blue);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    }else
    {
        g.setColor(Color.white);
        g.fillRect(x,150,25,400);
        g.setColor(Color.pink);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    }
}

```

```

        }
        x=x+40;
    }

    }
    public void update(Graphics g)
    {
        paint(g);
    }

void pause()
{

    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(600);} catch (InterruptedException e){}
}
void pause2()
{

    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(800);} catch (InterruptedException e){}
}

public void sort(int a[])
{
    for (;tmpi<a.length;tmpi++)
    {
        l1=tmpi;
        tmpda=a[tmpi];
        exg = false;
        for (tmpj=tmpi-1;tmpj>=0&&tmpda > a[tmpj];
tmpj--)
        {
            h1 = tmpj;
            pause2();

```

```

        counter = counter + 600;
        exg = true;
        count = count + 1;
        ecoun = ecoun + 1;
        a[tmpj + 1] = a[tmpj];
    }
    a[tmpj + 1] = tmpda;
    h1=l1-1;
        counter = counter + 600;
        count = count + 1;
        pause();
        if (step == true) {
            algsort.suspend();
        }
    }
    complete=true;
    repaint();
}

```

```

public void actionPerformed(ActionEvent e) {
    Object src = e.getSource();
    if( src==stop ) {

        run=false;
        algsort.suspend();
        return;
    }else if(src==start)
    {
        run=true;
        step=false;
        if (algsort == null)
        {
            algsort = new Thread(this);
            algsort.start();
        }else{
            algsort.resume();
        }

        return;
    }else if (src==one)

```

```

        {
            step=true;

            if (algsort == null)
                {
            algsort = new Thread(this);
            algsort.start();
            }else
            {
                algsort.resume();
            }
            return;
        }else if (src==ente)
        {
            ar[0]=100- Integer.parseInt(text0.getText());
            ar[1]=100- Integer.parseInt(text1.getText());
            ar[2]=100- Integer.parseInt(text2.getText());
            ar[3]=100- Integer.parseInt(text3.getText());
            ar[4]=100- Integer.parseInt(text4.getText());
            ar[5]=100- Integer.parseInt(text5.getText());
            ar[6]=100- Integer.parseInt(text6.getText());
            ar[7]=100- Integer.parseInt(text7.getText());
            ar[8]=100- Integer.parseInt(text8.getText());
            ar[9]=100- Integer.parseInt(text9.getText());
            repaint();
        }
    }
}

```

附錄 D Quick Sort 的 Applet 程式實作

Source Code

```

import java.awt.*;
import java.awt.event.*;
import java.io.InputStream;
import java.util.Hashtable;
import java.net.*;

```



```

import java.applet.*;

public class qusort extends java.applet.Applet implements
Runnable,MouseListener,ActionListener
{
    static final String message = "Quick Sort ";
    static final String message1 = "Total Time(ms):";
    static final String message2 = "Status:";
    static final String message3 = "To find the item which is smaller then
blue item";
    static final String message4 = "To find the item which is bigger then
blue item";
    static final String message6 = "Times of comparing:";
    static final String message7 = "Times of Exchange:";
    static final String message8 = "Sorting Complete!!!";
    static final String message11 = "Exchange!!!";
    static final String arrow = " ";
    static final String com1 = "is the pointer to find big one";
    static final String com2 = "is the pointer to find small one";
    static final String com3 = "is the compare one";
    static final String message9 = "please enter 10 number for sort";
    static final String message10 = "Do not bigger than 100";
    TextField text0=new TextField("28");
    TextField text1=new TextField("63");
    TextField text2=new TextField("79");
    TextField text3=new TextField("15");
    TextField text4=new TextField("44");
    TextField text5=new TextField("37");
    TextField text6=new TextField("27");
    TextField text7=new TextField("69");
    TextField text8=new TextField("18");
    TextField text9=new TextField("34");
    Button ente=new Button("Enter");

    String ti;
    String si;
    String ii;
    String s;
    private Font ffont;
    private Font fffont;
    private Font bffont;
    int h1=0;

```

```

int l1=0;
int m1=0;
int x=0;
    int k=0;
int seth=0;
int setl=0;
int tmpi=0;
int tmpj=0;
int counter=0;
int count=0;
    int[] ar=new int[ ]{100,100,100,100,100,100,100,100,100,100};
int ecount=0;
boolean exg=false;
boolean success=false;
boolean step=false;
boolean stopacc=false;
boolean run=true;
boolean warn=false;
boolean initial=true;
boolean complete=false;
boolean comps=false;
    boolean compb=false;
Button stop;
Button start;
Button one;

```

```

public void init()
{
    ffont=new Font ("arial",Font.BOLD,15);
    fffont=new Font ("arial",Font.BOLD,20);
        bffont=new Font ("arial",Font.BOLD,35);
setBackground(Color.white);
addMouseListener(this);
this.setLayout(null);
stop=new Button("Stop");
stop.setBounds(600,10,50,30);
this.add(stop);
stop.addActionListener(this);
start=new Button("Start");

```

```

start.setBounds(650,10,50,30);
this.add(start);
start.addActionListener(this);
one=new Button("Step by step");
one.setBounds(700,10,100,30);
this.add(one);
one.addActionListener(this);
    ente.setBounds(700,160,50,30);
    this.add(ente);
    ente.addActionListener(this);
    text0.setBounds(600,190,40,30);
    this.add(text0);
    text1.setBounds(650,190,40,30);
    this.add(text1);
    text2.setBounds(600,220,40,30);
    this.add(text2);
    text3.setBounds(650,220,40,30);
    this.add(text3);
    text4.setBounds(600,250,40,30);
    this.add(text4);
    text5.setBounds(650,250,40,30);
    this.add(text5);
    text6.setBounds(600,280,40,30);
    this.add(text6);
    text7.setBounds(650,280,40,30);
    this.add(text7);
    text8.setBounds(600,310,40,30);
    this.add(text8);
    text9.setBounds(650,310,40,30);
    this.add(text9);

}
Thread algsort=null;

public void start()
{
}
public void stop()
{
    algsort=null;
}

```

```

public void run()
{
    run=true;
    initial=false;
    sort(ar,0,9);
}
public void mouseClicked(MouseEvent e) {};
public void mouseReleased(MouseEvent e) {};
public void mouseEntered(MouseEvent e) {};
public void mouseExited(MouseEvent e) {};
public void mousePressed(MouseEvent e) {};

```

```

public void paint(Graphics g)
{
    int x=100;
    int y=100;
    g.setColor(Color.black);
    g.setFont(bfont);
    g.drawString(message,10,30);
    g.setFont(ffont);
    g.drawString(message1,10,50);
    g.setFont(ffont);
    g.drawString(message2,300,85);
    g.setFont(ffont);
    g.drawString(message6,10,70);
    g.setFont(ffont);
    g.drawString(message7,10,90);
        g.setFont(ffont);
    g.drawString(message9,550,130);
        g.setFont(ffont);
    g.drawString(message10,550,150);

    g.setFont(null);
        g.setColor(Color.black);
        g.drawString(arrow,300,25);
    g.setFont(ffont);
    g.setColor(Color.black);
    g.drawString(com1,320,25);

        g.setFont(null);

```

```

g.setColor(Color.red);
    g.drawString(arrow,300,45);
g.setFont(ffont);
g.setColor(Color.black);
g.drawString(com2,320,45);

        g.setColor(Color.blue);
        g.fillRect(300,55,15,15);
g.setColor(Color.black);
g.drawString(com3,320,65);

int tmp=0;
if(complete==true)
{
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message8,360,85);
        g.setColor(Color.black);
        g.setFont(ffont);
        g.drawString("Result:",500,370);
        for (int i=0;i<ar.length;i++)
        {
            k=100-ar[i];
            s = String.valueOf(k);
            g.drawString(s,500+30*i,390);
        }

        }else if(exg==true)
{
    g.setColor(Color.white);
    g.fillRect(360,70,500,20);
    g.setColor(Color.black);
    g.setFont(ffont);
    g.drawString(message11,360,85);

}else if(comps==true)
{

```

```

g.setColor(Color.white);
g.fillRect(360,70,500,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(message3,360,85);

    }else if(compb==true)
    {
        g.setColor(Color.white);
        g.fillRect(360,70,500,20);
        g.setColor(Color.black);
        g.setFont(f font);
        g.drawString(message4,360,85);
    }
ti=String.valueOf(counter);
g.setColor(Color.white);
g.fillRect(200,30,100,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(ti,200,50);
si=String.valueOf(count);
g.setColor(Color.white);
g.fillRect(200,50,100,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(si,200,70);
ii=String.valueOf(ccount);
g.setColor(Color.white);
g.fillRect(200,70,100,20);
g.setColor(Color.black);
g.setFont(f font);
g.drawString(ii,200,90);

    for(int i=0;i<ar.length;i++)
    {

        g.setColor(Color.white);
        g.fillRect(x,130,90,20);
        if (i==h1&&i==l1)
        {
            g.setColor(Color.white);

```

```

        g.fillRect(x,150,25,400);
        g.setColor(Color.pink);
        g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
        g.setFont(ffont);
        g.drawString(arrow,x,125+ar[i]*3);
        g.setColor(Color.red);
        g.setFont(ffont);
        g.drawString(arrow,x,145+ar[i]*3);
    }
else if (i==h1&&i==m1)
{
    g.setColor(Color.white);
    g.fillRect(x,150,25,400);
    g.setColor(Color.blue);
    g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.red);
    g.setFont(ffont);
    g.drawString(arrow,x,145+ar[i]*3);
}
else if (i==h1)
{
    g.setColor(Color.white);
    g.fillRect(x,150,25,400);
    g.setColor(Color.pink);
    g.fillRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.black);
    g.drawRect(x,150+ar[i]*3,20,300-ar[i]*3);
    g.setColor(Color.red);
    g.setFont(ffont);
    g.drawString(arrow,x,145+ar[i]*3);
}
else if(i==l1)
{
    g.setColor(Color.white);
    g.fillRect(x,150,25,400);
    g.setColor(Color.pink);

```

```

        g.fillRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
        g.setFont(ffont);
        g.drawString(arrow, x, 145+ar[i]*3);

    }else if (i==m1)
    {
        g.setColor(Color.white);
        g.fillRect(x, 150, 25, 400);
        g.setColor(Color.blue);
        g.fillRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
    }else
    {
        g.setColor(Color.white);
        g.fillRect(x, 150, 25, 400);
        g.setColor(Color.pink);
        g.fillRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
        g.setColor(Color.black);
        g.drawRect(x, 150+ar[i]*3, 20, 300-ar[i]*3);
    }
    x=x+40;
}

}

public void update(Graphics g)
{
    paint(g);
}

void pause()
{
    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(600);} catch (InterruptedException e){}
}

```



```

void pause2()
{

    if (algsort != null)
    {
        repaint();
    }
    try {Thread.sleep(800);} catch (InterruptedException e){}
}

public void sort(int a[],int fir,int fin)
{
    int fir_tmp, fin_tmp, tmp, flag;
    m1 = fir;
    if (fir < fin)
    {
        fir_tmp = fir + 1;
        l1 = fir_tmp;
        fin_tmp = fin;
        h1 = fin_tmp;
        tmp = a[fir];
        flag = 1;
        while (flag == 1)
        {
            exg=false;
            comps=false;
            compb=false;
            while (a[fir_tmp] > tmp)
            {
                compb=true;
                pause2();
                fir_tmp = fir_tmp + 1;
                l1 = fir_tmp;
                counter=counter+600;
                count=count+1;
                if(step==true)
                {
                    algsort.suspend();
                }
            }
            while (a[fin_tmp] < tmp)

```

```

{
    comps=true;
    pause2();
    fin_tmp = fin_tmp - 1;
    h1 = fin_tmp;
    counter=counter+600;
    count=count+1;
    if(step==true)
    {
        algsort.suspend();
    }
}
if (fir_tmp < fin_tmp)
{
    pause2();
    tmpi = a[fir_tmp];
    a[fir_tmp] = a[fin_tmp];
    a[fin_tmp] = tmpi;
    exg=true;
    counter=counter+600;
    ecount=ecount+1;
}
else {
    flag = 0;
}
counter=counter+600;
pause();
if(step==true)
{
    algsort.suspend();
}
}
pause2();
tmpj = a[fir];
a[fir] = a[fin_tmp];
a[fin_tmp] = tmpj;
ecount=ecount+1;
counter=counter+600;
sort(a, fir, fin_tmp - 1);

```

```

        sort(a, fin_tmp + 1, fin);
    }
    else if(l1==9)
    {
        complete = true;
        repaint();
    }
}
public void actionPerformed(ActionEvent e) {
Object src = e.getSource();
    if( src==stop ) {
run=false;
algsort.suspend();
return;
}else if(src==start)
{
    run=true;
    step=false;
    if (algsort == null)
    {
algsort = new Thread(this);
algsort.start();
}else{
algsort.resume();
    }
    return;
}else if (src==one)
{
    step=true;
    if (algsort == null)
    {
algsort = new Thread(this);
algsort.start();
}else
{
    algsort.resume();
}
return;
}else if (src==ente)
{
    ar[0]=100- Integer.parseInt(text0.getText());

```

```

        ar[1]=100-Integer.parseInt(text1.getText());
        ar[2]=100-Integer.parseInt(text2.getText());
        ar[3]=100-Integer.parseInt(text3.getText());
        ar[4]=100-Integer.parseInt(text4.getText());
        ar[5]=100-Integer.parseInt(text5.getText());
        ar[6]=100-Integer.parseInt(text6.getText());
        ar[7]=100-Integer.parseInt(text7.getText());
        ar[8]=100-Integer.parseInt(text8.getText());
        ar[9]=100-Integer.parseInt(text9.getText());
        repaint();
    }
}
}

```

附錄 E 測驗題的實作

```

#!/usr/bin/perl -w

print "content-type:text/html\n\n";
print "<html>";
print "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=big5\">";
print "<head>";
print "<title>Test html</title>";
print "</head>";
print "<body bgcolor=\"#FFFF99\">";
print "<H1>排序測驗</H1>";
print "<form method=\"post\" action=\"ans.pl\">";
$k=int(rand(15));
$i=0;
$z=$k;
for ($i=0;$i<4;$i++)
{
    $k=$k%15;
    if ($k==0)
    {
        print "<p>59,28,77,31,51,11 以插入排序法 排序由小到大 pass3 的結果  
應該是?</p>\n";
        print "<input type=\"hidden\" name=\"quest0\" value=\"1\">";
        print "<p><input type=\"radio\" name=\"radio0\" value=\"1\">";
    }
}

```

```

print "28,31,59,77,51,11</p>\n";
print "<p><input type=\"radio\" name=\"radio0\" value=\"2\">";
print "11,59,28,77,31,51</p>\n";
print "<p><input type=\"radio\" name=\"radio0\" value=\"3\">";
print "11,28,77,31,51,59</p>\n\n";
}
if ($k==1)
{
print "<p>59,28,77,31,51,11 以選擇排序法 排序由小到大 pass3 的結果
應該是?</p>\n";
print "<input type=\"hidden\" name=\"quest1\" value=\"1\">";
print "<p><input type=\"radio\" name=\"radio1\" value=\"1\">";
print "28,31,59,77,51,11</p>\n";
print "<p><input type=\"radio\" name=\"radio1\" value=\"2\">";
print "11,59,28,77,31,51</p>\n";
print "<p><input type=\"radio\" name=\"radio1\" value=\"3\">";
print "11,28,77,31,51,59</p>\n\n";
}
if ($k==2)
{
print "<p>59,28,77,31,51,11 以泡沫排序法 排序由小到大 pass3 的結果
應該是?</p>\n";
print "<input type=\"hidden\" name=\"quest2\" value=\"1\">";
print "<p><input type=\"radio\" name=\"radio2\" value=\"1\">";
print "28,31,59,77,51,11</p>\n";
print "<p><input type=\"radio\" name=\"radio2\" value=\"2\">";
print "11,59,28,77,31,51</p>\n";
print "<p><input type=\"radio\" name=\"radio2\" value=\"3\">";
print "11,28,77,31,51,59</p>\n\n";
}
if ($k==3)
{
print "<p>下列何種排序法的時間複雜度為  $O(n \log n)$ </p>\n";
print "<input type=\"hidden\" name=\"quest3\" value=\"1\">";
print "<p><input type=\"radio\" name=\"radio3\" value=\"1\">";
print "選擇排序法</p>\n";
print "<p><input type=\"radio\" name=\"radio3\" value=\"2\">";
print "快速排序法</p>\n";
}

```

```

print "<p><input type=\"radio\" name=\"radio3\" value=\"3\">";
print "泡沫排序法</p>\n\n";
}
if ($k==4)
{
print "<p>59,28,77,31,51,11 以快速排序法 排序並以第一筆資料為基準
則 pass1 的結果應該是?</p>\n";
print "<input type=\"hidden\" name=\"quest4\" value=\"1\">";
print "<p><input type=\"radio\" name=\"radio4\" value=\"1\">";
print "51,28,11,31,59,77</p>\n";
print "<p><input type=\"radio\" name=\"radio4\" value=\"2\">";
print "28,59,77,31,51,11</p>\n";
print "<p><input type=\"radio\" name=\"radio4\" value=\"3\">";
print "11,59,28,77,31,51</p>\n\n";
}
if ($k==5)
{
print "<p>在排序的過程中會逐一掃描所有的資料，發現資料所在的位
置錯誤，則會立即執行交換資料位置的動作。所以每一次掃描完畢後，至少會有一
筆資料被置於正確的位置。這是那一種排序法?</p>\n";
print "<input type=\"hidden\" name=\"quest5\" value=\"1\">";
print "<p><input type=\"radio\" name=\"radio5\" value=\"1\">";
print "泡沫排序法</p>\n";
print "<p><input type=\"radio\" name=\"radio5\" value=\"2\">";
print "插入排序法</p>\n";
print "<p><input type=\"radio\" name=\"radio5\" value=\"3\">";
print "快速排序法</p>\n\n";
}
if ($k==6)
{
print "<p>在排序的過程中會在所有資料中挑選最小的一筆資料項，置
於資料的第一個位置，然後在剩餘的資料中再找尋最小的一筆資料項，置於資料
的第二個位置，如此反覆執行，直到所有資料已由小到大排列。這是那一種排序
法?</p>\n";
print "<input type=\"hidden\" name=\"quest6\" value=\"1\">";
print "<p><input type=\"radio\" name=\"radio6\" value=\"1\">";
print "選擇排序法</p>\n";
print "<p><input type=\"radio\" name=\"radio6\" value=\"2\">";

```

```

print "插入排序法</p>\n";
print "<p><input type=\"radio\" name=\"radio6\" value=\"3\">";
print "快速排序法</p>\n\n";
}
if ($k==7)
{
    print "<p>排序方式為從陣列中依序取出一個元素，並檢查該元素值與陣列中已排序完成的資料項作比較，然後將該元素插入陣列中的適當位置。這是那一種排序法?</p>\n";
    print "<input type=\"hidden\" name=\"quest7\" value=\"1\">";
    print "<p><input type=\"radio\" name=\"radio7\" value=\"1\">";
    print "選擇排序法</p>\n";
    print "<p><input type=\"radio\" name=\"radio7\" value=\"2\">";
    print "泡沫排序法</p>\n";
    print "<p><input type=\"radio\" name=\"radio7\" value=\"3\">";
    print "插入排序法</p>\n\n";
}
if ($k==8)
{
    print "<p>N 個資料作氣泡排序時，須經過幾次比較？</p>\n";
    print "<input type=\"hidden\" name=\"quest8\" value=\"1\">";
    print "<p><input type=\"radio\" name=\"radio8\" value=\"1\">";
    print "N(N-1)/2</p>\n";
    print "<p><input type=\"radio\" name=\"radio8\" value=\"2\">";
    print "N/2</p>\n";
    print "<p><input type=\"radio\" name=\"radio8\" value=\"3\">";
    print "N(N+1)/2</p>\n\n";
}
if ($k==9)
{
    print "<p>逐次比較兩個相鄰的資料，若大小順序有誤，則交換位置，直到全部的資料置於正確的位置為止，以上為下列哪一種演算法？</p>\n";
    print "<input type=\"hidden\" name=\"quest9\" value=\"1\">";
    print "<p><input type=\"radio\" name=\"radio9\" value=\"1\">";
    print "泡沫排序法</p>\n";
    print "<p><input type=\"radio\" name=\"radio9\" value=\"2\">";
    print "選擇排序法</p>\n";
    print "<p><input type=\"radio\" name=\"radio9\" value=\"3\">";

```

```

    print "插入排序法</p>\n\n";
}
if ($k==10)
{
    print "<p>下列關於氣泡排序法的敘述中，哪一個是錯誤的？</p>\n";
    print "<input type=\"hidden\" name=\"quest10\" value=\"1\">";
    print "<p><input type=\"radio\" name=\"radio10\" value=\"1\">";
    print "在每一次的排列循環後，一定會有一個正確的資料會被置於正確
的位置</p>\n";
    print "<p><input type=\"radio\" name=\"radio10\" value=\"2\">";
    print "若排列 10 筆資料，最多須做 9 次的排列循環</p>\n";
    print "<p><input type=\"radio\" name=\"radio10\" value=\"3\">";
    print "適用於資料筆數很多的情況下</p>\n\n";
}
if ($k==11)
{
    print "<p>如果要將資料由小至大的順序排列，首先在所有的資料中選
擇最小的資料並和放在第一個位置上的資料交換位置，接著再從第二筆資料開始
選擇最小的資料，並和放在第二個位置上的資料交換位置，如此循環下去，直到
全部的資料都排序完成。這種排序法稱為？</p>\n";
    print "<input type=\"hidden\" name=\"quest11\" value=\"1\">";
    print "<p><input type=\"radio\" name=\"radio11\" value=\"1\">";
    print "泡沫排序法</p>\n";
    print "<p><input type=\"radio\" name=\"radio11\" value=\"2\">";
    print "選擇排序法</p>\n";
    print "<p><input type=\"radio\" name=\"radio11\" value=\"3\">";
    print "插入排序法</p>\n\n";
}
if ($k==12)
{
    print "<p>利用氣泡排序法排列 10 筆資料的順序，須經過幾次的比較？
</p>\n";
    print "<input type=\"hidden\" name=\"quest12\" value=\"1\">";
    print "<p><input type=\"radio\" name=\"radio12\" value=\"1\">";
    print "45 次</p>\n";
    print "<p><input type=\"radio\" name=\"radio12\" value=\"2\">";
    print "30 次</p>\n";
    print "<p><input type=\"radio\" name=\"radio12\" value=\"3\">";

```



```

        print "55 次</p>\n\n";
    }
    if ($k==13)
    {
        print "<p>利用插入排序法排列 10 筆資料的順序，最多做幾次的排序循環？</p>\n";
        print "<input type=\"hidden\" name=\"quest13\" value=\"1\">";
        print "<p><input type=\"radio\" name=\"radio13\" value=\"1\">";
        print "5 次</p>\n";
        print "<p><input type=\"radio\" name=\"radio13\" value=\"2\">";
        print "10 次</p>\n";
        print "<p><input type=\"radio\" name=\"radio13\" value=\"3\">";
        print "20 次</p>\n\n";
    }
    if ($k==14)
    {
        print "<p>以第一筆資料作為基準，將資料分為大於第一筆資料，與小於第一筆資料，兩個區塊，在分別對兩個區塊重複做相同的動作，此種方法稱為？</p>\n";
        print "<input type=\"hidden\" name=\"quest14\" value=\"1\">";
        print "<p><input type=\"radio\" name=\"radio14\" value=\"1\">";
        print "選擇排序法</p>\n";
        print "<p><input type=\"radio\" name=\"radio14\" value=\"2\">";
        print "插入排序法</p>\n";
        print "<p><input type=\"radio\" name=\"radio14\" value=\"3\">";
        print "快速排序法</p>\n\n";
    }
    $k=$k+2;
}
print "<input type=\"submit\" name=\"Submit\" value=\"OK\">";
print "</from>";
print "</body>";
print "</html>";

```

附錄 F 解答的實作

```
#!/usr/bin/perl -w
```

```
print "content-type:text/html\n\n";
```

```

print "<html>";
print "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=big5\">";
print "<head>";
print "<title>Test html</title>";
print "</head>";
print "<body bgcolor=\"#FFFF99\">";
print "<H1>排序解答</H1>";
&GetInput;
    if (${'quest0'}==1)
    {
        print "59,28,77,31,51,11 以插入排序法 排序由小到大 pass3 的結果應該是?<br>";
        print "28,31,59,77,51,11<br>";
        print "11,59,28,77,31,51<br>";
        print "11,28,77,31,51,59<br>";
        if (${'radio0'}==1)
        {
            print '恭喜答對囉!!!正確答案果然是 1<br><br>'
        }
        else
        {
            print '喔喔 抱歉..正確答案應該是 1<br><br>'
        }
    }
    if (${'quest1'}==1)
    {
        print "59,28,77,31,51,11 以選擇排序法 排序由小到大 pass3 的結果應該是?<br>";
        print "28,31,59,77,51,11<br>";
        print "11,59,28,77,31,51<br>";
        print "11,28,77,31,51,59<br>";
        if (${'radio1'}==3)
        {
            print '恭喜答對囉!!!正確答案果然是 3<br><br>'
        }
        else
        {

```

```

        print '喔喔 抱歉..正確答案應該是 3<br><br>'
    }

}
if ($in{'quest2'}==1)
{
    print "59,28,77,31,51,11 以泡沫排序法 排序由小到大 pass3 的結果應該
是?<br>";
    print "28,31,59,77,51,11<br>";
    print "11,59,28,77,31,51<br>";
    print "11,28,77,31,51,59<br>";
    if ($in{'radio2'}==2)
    {
        print '恭喜答對囉!!!正確答案果然是 2<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 2<br><br>'
    }
}
if ($in{'quest3'}==1)
{
    print "下列何種排序法的時間複雜度為  $O(n \log n)$ <br>";
    print "選擇排序法<br>";
    print "快速排序法<br>";
    print "泡沫排序法<br>";
    if ($in{'radio3'}==2)
    {
        print '恭喜答對囉!!!正確答案果然是 2<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 2<br><br>'
    }
}
if ($in{'quest4'}==1)

```

```

    {
        print "59,28,77,31,51,11 以快速排序法 排序並以第一筆資料為基準則
pass1 的結果應該是?<br>";
        print "51,28,11,31,59,77<br>";
        print "28,59,77,31,51,11<br>";
        print "11,59,28,77,31,51<br>";
        if ($in{'radio4'}==1)
        {
            print '恭喜答對囉!!!正確答案果然是 1<br><br>'
        }
        else
        {
            print '喔喔 抱歉..正確答案應該是 1<br><br>'
        }
    }
}
if ($in{'quest5'}==1)
{
    print '在排序的過程中會逐一掃描所有的資料，發現資料所在的位置錯誤，則會立即執行交換資料位置的動作。所以每一次掃描完畢後，至少會有一筆資料被置於正確的位置。這是那一種排序法?<br>';
    print "泡沫排序法<br>";
    print "插入排序法<br>";
    print "快速排序法<br>";
    if ($in{'radio5'}==1)
    {
        print '恭喜答對囉!!!正確答案果然是 1<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 1<br><br>'
    }
}
}
if ($in{'quest6'}==1)
{
    print '在排序的過程中會在所有資料中挑選最小的一筆資料項，置於資料的第一個位置，然後在剩餘的資料中再找尋最小的一筆資料項，置於資料的第

```

二個位置，如此反覆執行，直到所有資料已由小到大排列。這是那一種排序法?

";

```
print "選擇排序法<br>";
print "插入排序法<br>";
print "快速排序法<br>";
if ($in{'radio6'}==1)
{
    print "恭喜答對囉!!!正確答案果然是 1<br><br>"
}
else
{
    print "喔喔 抱歉..正確答案應該是 1<br><br>"
}
```

```
}
if ($in{'quest7'}==1)
{
    print "排序方式為從陣列中依序取出一個元素，並檢查該元素值與陣列中已排序完成的資料項作比較，然後將該元素插入陣列中的適當位置。這是那一種排序法?  
<br>";
```

```
print "選擇排序法<br>";
print "泡沫排序法<br>";
print "插入排序法<br>";
if ($in{'radio7'}==3)
{
    print "恭喜答對囉!!!正確答案果然是 3<br><br>"
}
else
{
    print "喔喔 抱歉..正確答案應該是 3<br><br>"
}
```

```
}
if ($in{'quest8'}==1)
{
    print "N 個資料作氣泡排序時，須經過幾次比較？<br>";
    print "N(N-1)/2<br>";
    print "N/2<br>";
```

```

print "N(N+1)/2<br>";
if ($in{'radio8'}==1)
{
    print '恭喜答對囉!!!正確答案果然是 1<br><br>'
}
else
{
    print '喔喔 抱歉..正確答案應該是 1<br><br>'
}

}
if ($in{'quest9'}==1)
{
    print "逐次比較兩個相鄰的資料，若大小順序有誤，則交換位置，直到
全部的資料置於正確的位置為止，以上為下列哪一種演算法？<br>";
    print "泡沫排序法<br>";
    print "選擇排序法<br>";
    print "插入排序法<br>";
    if ($in{'radio9'}==1)
    {
        print '恭喜答對囉!!!正確答案果然是 1<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 1<br><br>'
    }

}
if ($in{'quest10'}==1)
{
    print "下列關於氣泡排序法的敘述中，哪一個是錯誤的？<br>";
    print "在每一次的排列循環後，一定會有一個正確的資料會被置於正確
的位置<br>";
    print "若排列 10 筆資料，最多須做 9 次的排列循環<br>";
    print "適用於資料筆數很多的情況下<br>";
    if ($in{'radio10'}==3)
    {
        print '恭喜答對囉!!!正確答案果然是 3<br><br>'
    }
}

```

```

    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 3<br><br>'
    }

}
if (${'quest11'}==1)
{
    print '如果要將資料由小至大的順序排列，首先在所有的資料中選擇最小的資料並和放在第一個位置上的資料交換位置，接著再從第二筆資料開始選擇最小的資料，並和放在第二個位置上的資料交換位置，如此循環下去，直到全部的資料都排序完成。這種排序法稱為?<br>';
    print '泡沫排序法<br>';
    print '選擇排序法<br>';
    print '插入排序法<br>';
    if (${'radio11'}==2)
    {
        print '恭喜答對囉!!!正確答案果然是 2<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 2<br><br>'
    }

}
if (${'quest12'}==1)
{
    print '利用氣泡排序法排列 10 筆資料的順序，須經過幾次的比較？<br>';
    print '45 次<br>';
    print '30 次<br>';
    print '55 次<br>';
    if (${'radio12'}==1)
    {
        print '恭喜答對囉!!!正確答案果然是 1<br><br>'
    }
    else

```

```

    {
        print '喔喔 抱歉..正確答案應該是 1<br><br>'
    }

}
if ($in{'quest13'}==1)
{
    print "利用插入排序法排列 10 筆資料的順序，最多做幾次的排序循環？
<br>";
    print "5 次<br>";
    print "10 次<br>";
    print "20 次<br>";
    if ($in{'radio13'}==2)
    {
        print '恭喜答對囉!!!正確答案果然是 2<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 2<br><br>'
    }

}
if ($in{'quest14'}==1)
{
    print "以第一筆資料作為基準，將資料分為大於第一筆資料，與小於第
一筆資料，兩個區塊，在分別對兩個區塊重複做相同的動作，此種方法稱為?<br>";
    print "選擇排序法<br>";
    print "插入排序法<br>";
    print "快速排序法<br>";
    if ($in{'radio14'}==3)
    {
        print '恭喜答對囉!!!正確答案果然是 3<br><br>'
    }
    else
    {
        print '喔喔 抱歉..正確答案應該是 3<br><br>'
    }
}

```



```

    }
print "</body>";
print "</html>";
sub GetInput
{
    my($method) = $ENV{'REQUEST_METHOD'};
    local($query, @in, $key, $val);
    if ($method eq 'GET') {
        $query = $ENV{'QUERY_STRING'};
    }
    elsif ($method eq 'POST') {
        read(STDIN, $query, $ENV{'CONTENT_LENGTH'});
    }
    local(@query) = split(/&/, $query);
    foreach (@query) {
        tr/+/ /;
        ($key, $val) = split(/=/);
        $key =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack("c", hex($1))/ge;
        $val =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack("c", hex($1))/ge;
        $val =~ s/\r\n/\n/g;
        $in{$key} = $val;
    }
}

```