

110年專門職業及技術人員高等考試建築師、
24類科技師（含第二次食品技師）、大地工程技師
考試分階段考試（第二階段考試）、公共衛生師
考試暨普通考試不動產經紀人、記帳士考試試題

代號：01360
頁次：5-1

等 別：高等考試

類 科：資訊技師

科 目：程式設計

考試時間：2 小時

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

- 一、假設有一家新成立不久的 L 公司，欲進行員工基本資料的建立。其基本資料包括有 ID（所有員工一旦被招募進入公司的第一天，就會給予一個 ID（數字））、姓（Last name）、名（First name）、年齡和月薪。請利用 Java 或 C++ 實作一完整程式，可以在命令行（command line）依序傳入一系列的所有員工基本資料，並將其存入到一個文字檔中（例如“salary.txt”）。最後，也能夠求算印出當月所有員工薪水加總之後的總平均、最低薪水和最高薪水（三者均為浮點數，取到小數第一位）。請適當地加入例外處理，讓使用者能夠依其需求，連續地在控制台輸入資料。亦即若其中有資料格式不符，可以印出例如“Invalid input! Try again!”或“輸入資料格式不符，請重新輸入!”的警示語，隨後程式還可以繼續進行到所有的資料輸入正確為止（例如使用者可以到最後打入結束指標（end-of-file indicator）結束繼續輸入）。一旦輸入結束，即提示或告知檔案已關閉（File Close），並進行「總員工人數」、「總平均薪水」、「最低薪水」和「最高薪水」的印出顯示。以下是利用 Java 實作時的執行範例。請使用 Java 或 C++ 的語法撰寫程式碼。（20 分）

假如未有任何的輸入：

```
Enter Employee ID, Last name, First name, and Monthly-salary
(Enter end-of-file indicator(CTRL+Z) to end input.)
?
close file
Silly! Nothing input!
```

正確輸入及最後執行的結果：

```
Enter Employee ID, Last name, First name, and Monthly-salary
(Enter end-of-file indicator(CTRL+Z) to end input.)
? 羅 ○華 56789
Invalid input.Try again!
? 1 羅 ○華 56789
? 2 林 ○曾 3456a
Invalid input.Try again!
? 2 林 ○曾 34567
? 3 劉 ○俊 23488
? 4 林 ○君 56789
? 5 方 ○滿 55555
?
close file
There are 5 employees considered
Average Salary in this month: 45437.6
Highest Salary in this month: 56789.0
Lowest Salary in this month: 23488.0
```

存到 “salary.txt” 裡的內容為

```
1 羅 ○華 56789.0
2 林 ○曾 34567.0
3 劉 ○俊 23488.0
4 林 ○君 56789.0
5 方 ○滿 55555.0
```

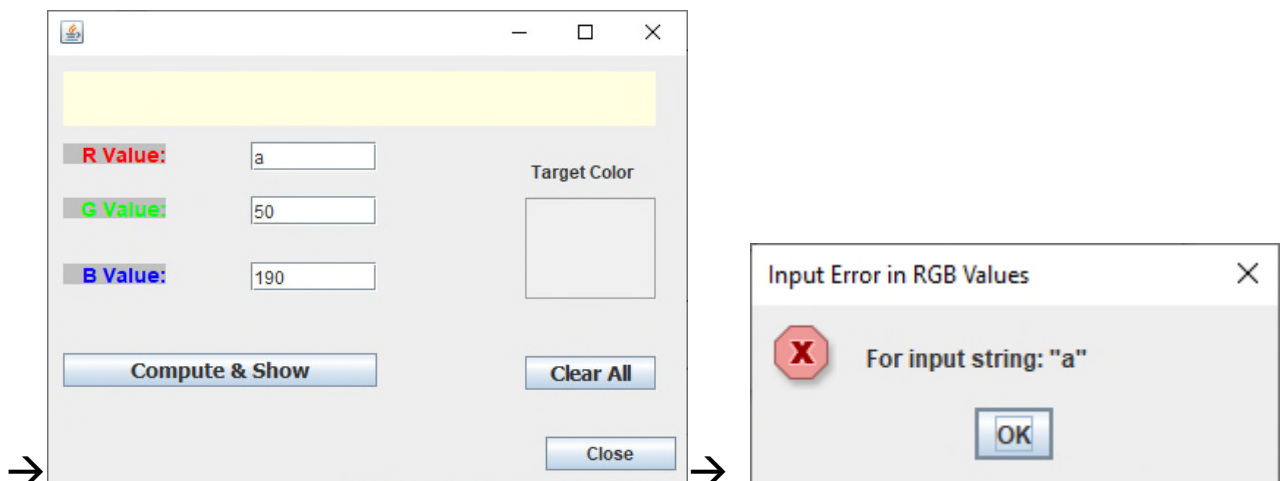
二、假設臺灣 XX 食品技師協會欲針對各類「生鮮食材新鮮度」的判定能夠依據較科學性的方法，乃打算研發出一套能夠應用「色彩計畫」進行食材「自然顏色與光澤外觀」和「新鮮度」關係等級的對照表資料庫。為達到此目標，乃擬定先期先發展出一套色彩轉換雛型系統程式。此系統得符合下列幾項基本的使用要求和程序：

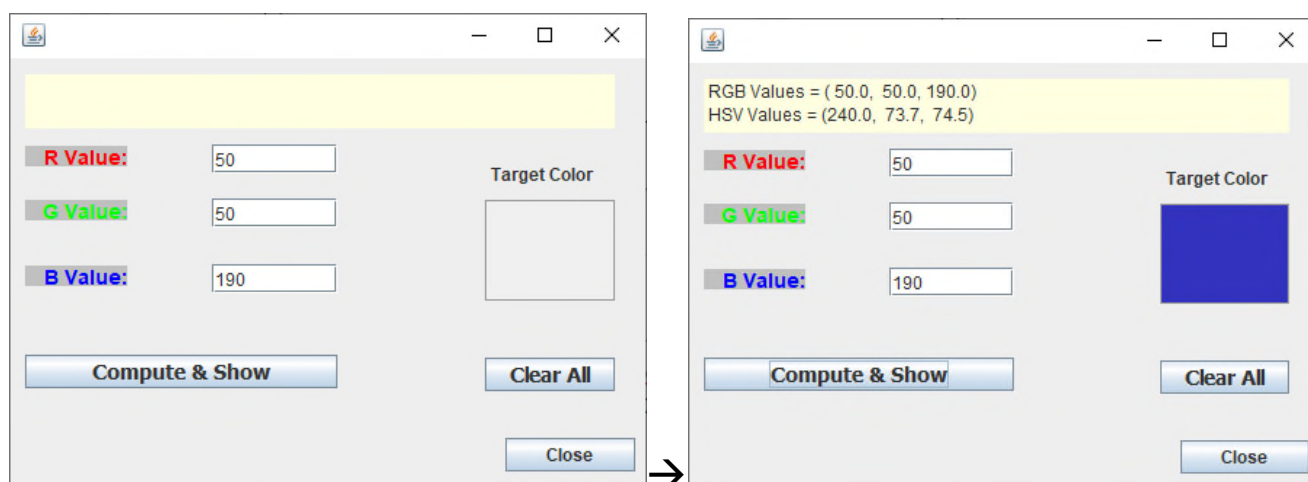
1. 輸入所收集到一「用相機所拍攝到的食材的 RGB 值」（所使用的相機已被準確校正，其所拍攝到 RGB 即代表真實色）；
2. RGB 的 3 個頻道值範圍均在 0~255 之間。
3. 為了讓此系統的使用者能對色澤外觀有較直觀的感覺和判別，得將 RGB 值轉成和「人眼視覺三屬性（Hue 色相、Saturation 飽和度、Value 明暗度）相關」的 HSV 色彩空間的表示法。
4. 可以輸入每筆建築相關顏色 RGB；隨後，即進行計算轉換成 HSV 值後，再將相關的 RGB 和 HSV 值輸出顯示。

假設你就是此套系統的雛型開發師，請利用 Java 或 C++ 撰寫一個以「圖形使用者介面」（Graphical User Interface, GUI）設計，並以 RGBbase 為超類別（Superclass）和 HSV_ColorSpace 為繼承的子類別之系統程式。（30 分）

以下是利用 Java 和 JFrame 實作時的執行範例。

- 提示：1. 在此作答中，RGB 轉換成 HSV 的方法（Method）或函式（Function）不需要實踐，僅以黑箱的方式處理即可，例如：
- ```
Public void RGB_to_HSV (double rValue, double gValue, double bValue) {...}
```
2. Target Color 的顯示以及其他元件（例如 labels 文字）全部就用 Default 的顏色（或文字字型 and 大小）設定，在此只是用來直觀視覺式的 Demo 輔助，你不需要實踐程式碼。





三、請使用 Java 或 C++語法撰寫「Switch 敘述流程控制」為主的水果超商計價程式碼。這個程式會列出四種（例如蘋果、木瓜、香蕉和西瓜）所銷售的水果名稱做選擇，並同時要求輸入其量秤所得到的磅重。每輸入一筆資料（水果名稱代碼和量秤磅重）後，即會印出該筆售價。隨後會再要求使用者繼續輸入下一筆不同的水果資料，直到此相關購買資料全部輸入完成。最後，再列出該交易的總售價金額。以下是利用 Java 實作時的執行範例。（25 分）

```
Type the end-of-file indicator to terminate input:
On UNIX/Linux/macOS type <Ctrl> d then press Enter
On Windows type <Ctrl> z then press Enters
=====
1)Enter the fruit type Chosen (integer values in the range 1-4) AND 2)Pounds Weighted
1:Apple(20/pound) 2:Banana(10/pound) 3:Papaya(30/pound) 4:Watermelon(10/pound)
?
1 10
Apple price:200.0

1)Enter the fruit type Chosen (integer values in the range 1-4) AND 2)Pounds Weighted
1:Apple(20/pound) 2:Banana(10/pound) 3:Papaya(30/pound) 4:Watermelon(10/pound)
? 3 20
Papaya price:600.0

1)Enter the fruit type Chosen (integer values in the range 1-4) AND 2)Pounds Weighted
1:Apple(20/pound) 2:Banana(10/pound) 3:Papaya(30/pound) 4:Watermelon(10/pound)
? 2 5
Banana price:50.0

1)Enter the fruit type Chosen (integer values in the range 1-4) AND 2)Pounds Weighted
1:Apple(20/pound) 2:Banana(10/pound) 3:Papaya(30/pound) 4:Watermelon(10/pound)
? 4 10
Watermelon price:100.0

1)Enter the fruit type Chosen (integer values in the range 1-4) AND 2)Pounds Weighted
1:Apple(20/pound) 2:Banana(10/pound) 3:Papaya(30/pound) 4:Watermelon(10/pound)
?
Selling Report:
Total selling price is 950
```

四、請依據下列 Java 程式碼執行後，回答輸出結果為何？（25 分）

```
public class CollectionArrayList4Colors {
 public static void main(String[] args) {
 String[] colors = { "RED", "Green","BLUE", "CYAN", "MAGENTA", "YELLOW","WHITE", "GRAY",
 "BLACK"};
 List<String> list = new ArrayList<String>();

 for (String color:colors) {
 list.add(color);
 }

 System.out.println("ArrayList: ");
 for (int count = 0; count < list.size(); count++) {
 System.out.printf("%s ", list.get(count));
 }

 String[] removeNeutrals = {"GRAY", "BLACK", "WHITE"};
 List<String> removeList = new ArrayList<String>();
 for (String color : removeNeutrals) {
 removeList.add(color);
 }

 removeColors(list, removeList);
 convertToLowercaseStrings(list);

 System.out.printf("\n\nArrayList after converting to Lowercase and calling
 removeColors:%n");
 for (String color:list) {
 System.out.printf("%s ", color);
 }
 }
}

private static void removeColors(Collection<String> collection1,
 Collection<String> collection2) {
 Iterator<String> iterator = collection1.iterator();

 while (iterator.hasNext()) {
 if (collection2.contains(iterator.next())) {
 iterator.remove();
 }
 }
}

private static void convertToLowercaseStrings(List<String> list) {
 ListIterator<String> iterator = list.listIterator();

 while (iterator.hasNext()) {
 String color = iterator.next();
 iterator.set(color.toLowerCase());
 }
}

//end class CollectionArrayList4Colors
```