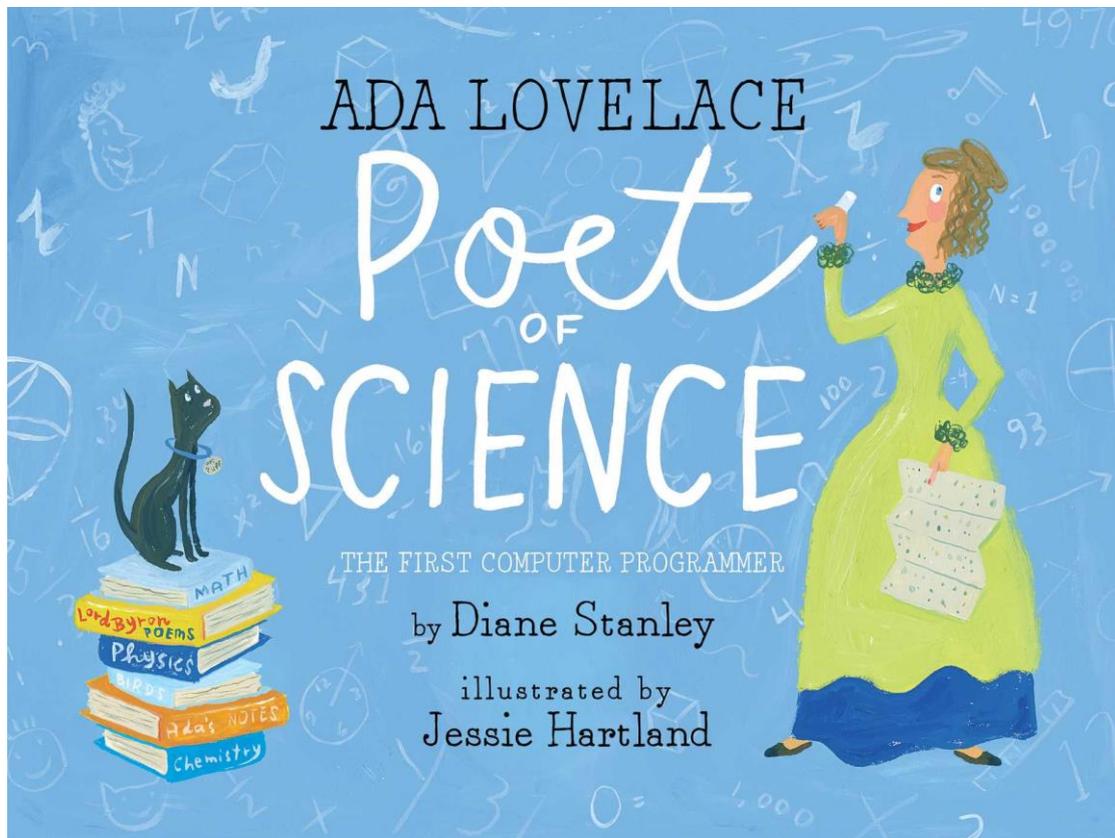


臺北市立第一女子高級中學 107 學年度  
資訊學科能力競賽初賽



程式設計試題 2018.05.15

答題注意事項：

1. 請留意題目的說明，題目中未提及的事項，請勿自行假設。例如，題目要求輸入一個整數時，勿假設此整數必為正整數。
2. 評分時的測試資料由評分老師提供，題目上的輸入輸出範例，視同題意的說明，並非評分的依據。輸出格式需完全和題目規定相同，不可以輸出多餘空白。
3. 程式內不可以含有 `system("pause")` 等等系統函數，如因此而無法評分，由參賽者自行負責。
4. 程式編寫時間：09:10 ~ 11:50 計分板凍結時間：11:20
5. 程式測試時間：11:50 ~ 12:00
6. 請填寫選手基本資料為後續培訓與競賽報名聯繫之用。  
<https://goo.gl/forms/GKAsKUQbmP5rToB11>

## P1. 青蛙王子去旅行 (檔案名稱: P1.CPP)

(陣列、迴圈★) 改自 d660: 11764 - Jumping Mario

蛙蛙原本是一個帥氣的王子，卻因為愛上公主而受到巫婆的詛咒。巫婆所設下的條件，就是要他到各個地方去旅行，蒐集隱藏的物品，才能恢復成王子的樣子。今天，風和日麗，青蛙王子又要出發去旅行了。他經過一片具有神奇力量的荷塘，蝸牛、蜜蜂、烏龜告訴他，只要通過高高低低的荷葉彈跳挑戰，就可以獲得三葉草，換取隱藏物品，並且購買明信片寄給守候著他的公主。

1

現在，請幫助蛙蛙統計他每一回合彈跳挑戰可以收集到的三葉草數量。

**a high jump**: 跳躍高度比前次高，可以收集 10 倍高度差的三葉草

**a low jump**: 跳躍高度比前次低，只能可以收集 2 倍高度差的三葉草

蛙蛙從地面起跳，最後跳回地面，高度皆為 0。

若 5 片荷葉高度分別為 1 2 3 4 3，則共有 4 次 high jump，2 次 low jump，可收集三葉草數量為  $(1-0) * 10 + (2-1) * 10 + (3-2) * 10 + (4-3) * 10 + (4-3) * 2 + (3-0) * 2 = 48$  片

### 輸入說明

第一行輸入的是一個整數  $T$  ( $T \leq 30$ )，表示接下來有  $T$  回合彈跳挑戰。

每筆測資開始於一個整數  $N$  ( $N \leq 30$ )，表示接下來有  $N$  片荷葉。

下一行依序為蛙蛙跳過  $N$  片荷葉的高度。

### 輸出說明

先輸出回合數，再輸出收集的三葉草數量。

格式請參考範例輸出 (請留意輸出文字的大小寫與冒號後的空白)

### 範例輸入

```
4
5
1 2 3 4 3
8
1 4 2 2 3 5 3 4
1
9
5
10 20 30 40 25
```



### 範例輸出

```
Round 1: 48
Round 2: 96
Round 3: 108
Round 4: 480
```



## P2. 駭客不要來 - 密碼分析器 (檔案名稱: P2.CPP)

(字元、字串處理☆☆) 改自 a624: 4. Password Analyzer

線上購物、網路交易等電子商務形式已成為現代人的日常，「密碼」是大家保障資訊安全的第一道防線。然而，多數人為圖方便，常使用簡短或常見的英文單字作為密碼，使得駭客常得以輕易猜測密碼，進而盜用使用者帳號。網路安全公司日前公佈最容易被駭客破解的密碼前五名為：第一名：123456；第二名：1234；第三名：123456789；第四名：password；第五名：iloveyou。

密碼怎麼設其實有很大的學問。如果使用者選擇較好的密碼，電子商務公司就可以省點力氣來處理駭客盜用帳號的問題。例如，增加長度或混用字母、數字及符號，都可以增加密碼猜測的難度，也同時增加密碼強度，提高安全性。

現在，聰明的妳，被賦予的任務是設計一個密碼分析器，提醒使用者選用密碼的強度。以下是密碼強度分析說明：

「強」(STRONG) 密碼必須同時擁有以下所有特性：

- 長度至少 8 個字元 (如："spookyfish")
- 同時有大寫及小寫字母 (如："sPookyFISH")
- 同時有字母及至少一個數字或符號 (如："sPookyFiSH3" 或 "\$PookyFI3H")

「好」(GOOD) 密碼：擁有兩個上述的特性；

「尚可」(ACCEPTABLE) 密碼：只擁有一個特性；

「弱」(WEAK) 密碼：不符合上述任何特性。

### 輸入說明

每筆測資多行，每行有一個密碼，長度最大為 30 字元。密碼中沒有空白。

### 輸出說明

請參考範例輸出密碼的強度。(請留意輸出文字的大小寫)

### 範例輸入

```
lizard
aardvark
Aardvark
Aardvark77
```

### 範例輸出

```
This password is WEAK
This password is ACCEPTABLE
This password is GOOD
This password is STRONG
```



### P3. 小綠綠的期中專題-影像亮度分析 (檔案名稱: P3.CPP)

(結構、二維陣列、排序、迴圈 ★★) 改自 b460: 判若黑白

小綠綠有一份影像處理期中專題報告，老師希望她們可以對 PPM 影像進行分析，並提出任何可能的影像處理方案。進行專題作業前，老師提醒大家，適度的掌握影像統計資訊，有助於色調校正與變化的判斷。為了讓小綠綠順利完成後續專題探討，現在就請妳一起陪著她進行基礎影像分析，了解影像亮度分佈情況。

3



PPM (Portable PixMap) 是一種可以將影像中每個像素之紅光 (Red)、綠光 (Green) 與藍光 (Blue) 三個亮度值分別記錄下來的格式。

影像寬: Width

影像高: Height

影像像素總量  $N = \text{Width} * \text{Height}$  (整數運算)

影像大小 (bytes):  $\text{Size} = \text{Width} * \text{Height} * 3$  (整數運算)

影像中的任意像素點亮度值  $P_{i,j} = (R_{i,j} + G_{i,j} + B_{i,j}) / 3$  (整數運算)

整張影像的平均亮度值  $\text{Mean} = \sum_{i,j}^{height,width} P_{i,j} / N$  (整數運算)

整張影像亮度值差異 (標準差)  $\text{SD} = \sqrt{\frac{1}{N} \sum_{i,j}^{height,width} (P_{i,j} - \text{Mean})^2}$

整張影像亮度的中間值 Mid: 將 N 個 pixel 亮度值排序後，最中間位置的亮度值。(若 N 為奇數，取第  $N/2$  個亮度值；若 N 為偶數，取第  $(N-1)/2$  與第  $N/2$  個亮度值之平均) (整數運算)



### 輸入說明

第一行有兩個整數 W、H ( $1 \leq W, H \leq 256$ ) 分別代表影像的寬(W)與高(H)。  
接下來有 H 行，每行有  $W \times 3$  個整數，是每個像素的 RGB 值 ( $0 \leq R, G, B \leq 255$ )。

### 輸出說明

輸出有 4 行。分別為  
影像大小(Size)  
平均亮度值(Mean，輸出至整數位)  
亮度值標準差(SD，輸出至小數後 2 位)  
亮度中間值(Mid，輸出至整數位)  
輸出格式請參考範例輸出 (請留意等號前後的空白與文字大小寫)

### 範例輸入

```
5 6
0 0 0 255 255 255 0 0 0 255 255 255 0 0 0
255 255 255 0 0 0 255 255 255 0 0 0 255 255 255
0 0 0 255 255 255 0 0 0 255 255 255 0 0 0
255 255 255 0 0 0 255 255 255 0 0 0 255 255 255
0 0 0 255 255 255 0 0 0 255 255 255 0 0 0
255 255 255 0 0 0 255 255 255 0 0 0 255 255 255
```

### 範例輸出

```
SIZE = 90 Bytes
MEAN = 127
SD = 127.50
MID = 127
```



## P4. 成雙成對-算式檢查 (檔案名稱：P4.CPP)

(字串、堆疊 ★★) 改自 b304: 00673 - Parentheses Balance

阿鵬正在學程式設計，興奮地幫阿希設計了一個數學作業計算器，只要把數學算式輸入進去，就可以立刻算出答案。但是迷糊又天然呆的阿希，不知道運算式加括號時左右必須成雙成對，因此，常常在輸入算式時不是少打一個括號，就是多打括號，甚致，有時還會左右不分，讓程式輸不出結果，久久沒有回應，阿希一直在偷偷抱怨阿鵬設計的程式笨笨的。

妳也學了一段時間的程式設計了，來幫阿鵬的計算器變得更聰明。加上一段成雙成對運算式檢查功能，可以先檢驗阿希輸入的運算式，如果是括號出現不合法，就說 Try again，請阿希重新輸入；如果合法，就說 Valid，才接著阿鵬的程式開始進行運算。運算式字串的最大長度為 128 個字元。

阿希的輸入值是一個運算式字串。以下為合法的情境：

1. 運算式輸入過程允許空白，例如： $[(3 + 6) * 4 - 2] =$
2. 括號中若為一個空字串也是正確的運算式，例如： $()$
3.  $()$  和  $[]$  都是括號的功能但必須成對出現，像是： $() ()$ ， $(( ))$ ， $([])$ ， $((([()])[]))$ ， $[][]([[]])$
4. 如果 A 和 B 都為正確的運算式，則 AB 也為正確的運算式
5. 如果 A 為正確的運算式，則  $(A)$  及  $[B]$  都為正確的運算式

### 輸入說明

輸入的第一列為正整數 n，代表接下來有 n 列待測運算式。

### 輸出說明

檢查每列待測運算式，如果正確輸出 Valid，否則輸出 Try again。(請留意輸出文字的大小寫)

### 範例輸入

```
3
[(3 + 6) * 4 - 2] =
[[([4*(3+5)])*2)+5]] =
([(3+5)*[4+6]/(5*8)]*2)-(5 ^ 2) =
```

### 範例輸出

```
Valid
Try again
Valid
```



## P5. 營運分析師 (檔案名稱: P5.CPP)

(結構、排序、基本運算★☆☆) 改自 b119: 售票系統 (Sales)

低利航空公司指的是將營運成本控制至比傳統航空公司為低的航空公司。低利航空公司因與同業競爭，經常給予預購票大幅折扣。近年來因為油價高漲，經過細算，該公司所飛行的航班必須以票面價售出 30% 的機位才能打平所有的支出，因此該公司必須減少每張票的折扣數或者提高票面單價，來提昇公司獲利水準。

但是全面提高票價可能會造成客源的流失，因此企畫部門提出了以下的動態調整機位售價的模式：給定某航段的票面價及機位數，前 20% 賣出的機位售價為票面價之 70%，前 21%~40% 賣出的機位售價為票面價之 80%，前 41%~60% 賣出的機位售價為票面價之 90%，最後售出的機位則全部為票面價。

作為一個營運分析師，請寫一個程式來幫助「低利航空公司」計算每日各飛航班機的利潤或虧損。計算過程皆四捨五入至整數。

### 輸入說明

第一行為本日航班數  $F$ ， $0 < F \leq 10$

接下來有  $F$  行航班資訊，每筆航班資訊內容以空白隔開。

分別代表班機編號  $fn$  (編號字串長度小於 10 位)、航班的票面價  $s$  ( $s \leq 5000$ )，機位數  $t$  ( $t \leq 250$ )，以及  $n$  個訂位記錄 ( $0 < n \leq 10$ )，後面則有  $n$  個整數，分別代表每個訂位的機位數。當然，所訂的機位總數不會超過  $t$ 。

### 輸出說明

請以各班機利潤高低 (由大到小) 排序，依次輸出各航班班機號碼與利潤 (或虧損)

### 範例輸入

```
2
BA-1023 2000 60 6 5 10 20 14 1 5
AA-3058 5000 240 5 3 14 14 14 10
```

### 範例輸出

```
BA-1023 59600
AA-3058 -164000
```

