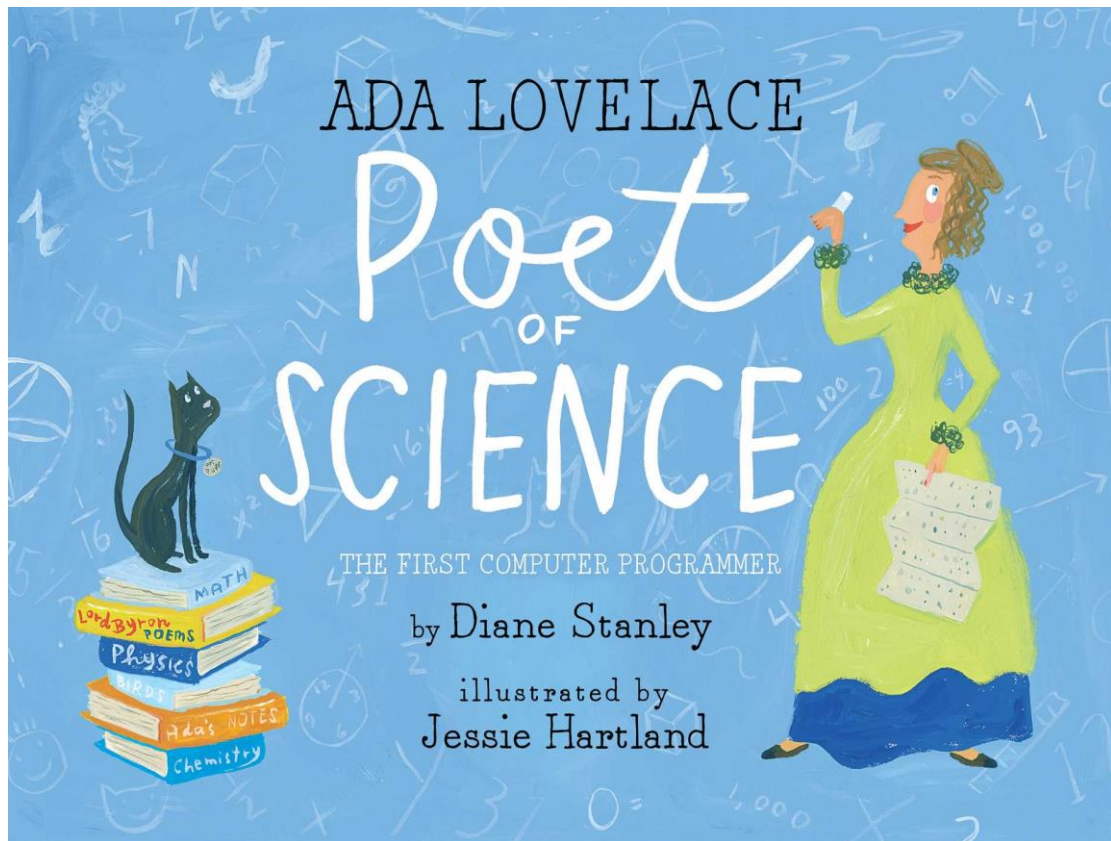


臺北市立第一女子高級中學 107 學年度
資訊學科能力競賽複賽



程式設計試題 2018.09.11

答題注意事項：

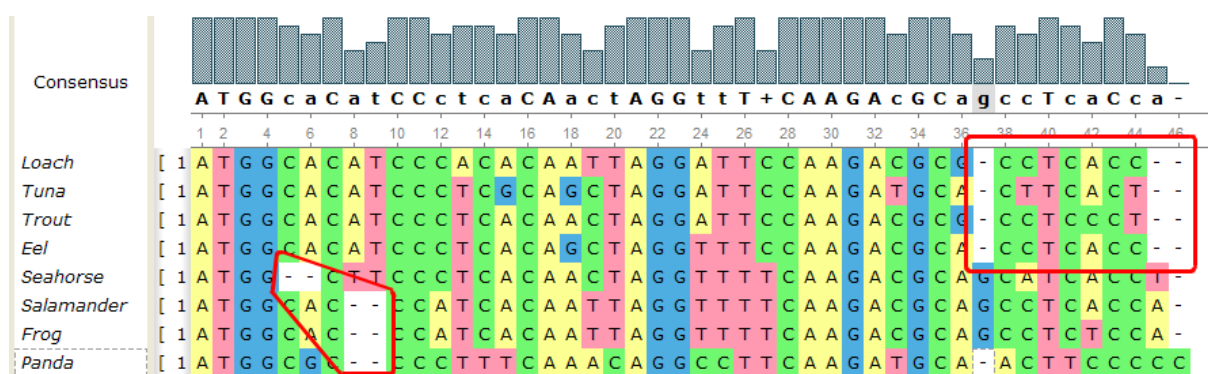
1. 請留意題目的說明，題目中未提及的事項，請勿自行假設。例如，題目要求輸入一個整數時，勿假設此整數必為正整數。
2. 評分時的測試資料由評分老師提供，題目上的輸入輸出範例，視同題意的說明，並非評分的依據。
3. 輸出格式需完全和題目規定相同，不可以輸出多餘空白。
4. 程式內不可以含有 `system("pause")` 等等系統函數，如因此而無法評分，由參賽者自行負責。
5. 程式編寫時間：09:10 ~ 11:50 計分板凍結時間：11:20
6. 程式測試時間：11:50 ~ 12:10

P1. 生物資訊：基因序列比對 (P1.CPP)

(動態規劃★★★)

隨著 DNA 定序技術的不斷提升，各種基因體計畫發展，越來越多的生物序列資料被建構成資料庫，並置於網際網路上免費供人使用。例如存放核酸序列資料的 GenBank、EMBL-Bank 和 DDBJ 等，以及存放蛋白質序列資料的 Swiss-Prot、TrEMBL 和 PIR 等。據估計，平均每 15 個月這些資料庫的內容就會增加一倍。利用這些資料，生物學家再藉由生物資訊和計算生物的分析工具，便可以了解許多生物問題。

在諸多生物資訊和計算生物的分析工具中，序列比對 (sequence alignment) 是一個基本且相當重要的研究工具，它可以比較及分析出兩條或多條序列之間的相似程度。相似度高的序列彼此間會有相似的結構及功能，這意味著它們可能源自共同的祖先。因此，生物學家一旦拿到未知功能的 DNA 或蛋白質序列時，最常做的事情就是利用序列比對工具搜尋資料庫，看看是否有已知註解功能的序列與手中未知功能的序列相似者，藉此推測手中序列的生物功能。



(資料來源：<https://ugene.net/wiki/display/UUOUM/Aligning+Sequences+to+a+Profile+with+MUSCLE>)

通常生物學家會比較喜歡利用比對來衡量兩條序列之間的相似程度。拿 GACGGATAG 和 GATCGGAATAG 這兩條 DNA 序列來說，乍看之下這兩條長度不同的 DNA 序列似乎不太相似。但是，當我們把它們重疊在一起，並在第一條序列的第二個和第三個字母之間與第六個和第七個字母之間分別插入一個空白字，就可發現其實這兩條 DNA 序列還蠻相像的。這種序列重疊的方式，就稱為序列的比對。

GACGGATAG
GATCGGAATAG

➡

GA CGGA TAG
GATCGGAATAG

過去這種基因研究的工作，生物學家得純靠手工進行序列資料庫的比對和搜尋，通常得花費數年才能完成。現在利用電腦比對搜尋，可能只需幾秒鐘而已。如何設計出有效率的演算法來解決兩條序列比對的問題呢？當要對齊的序列長度還算小的情況下，我們可以採用暴力法或窮舉法來解決這個問題：就是列舉出所有的對齊方式，然後再從中找出得分最高的對齊方式。但是，若要對齊的序列長度很大，這種暴力或窮舉的方法就行不通了！因為單單要列舉出所有的對齊方式，本身就是一件困難的事情。



假設要對齊的兩條序列長度都是 n ，那麼它們之間總共就有 $\binom{2n}{n} \approx \frac{2^{2n}}{\sqrt{\pi n}}$ 個對齊方式。換句話說，對齊方式的個數會隨著序列的長度呈指數成長。

以人類整個基因體來說，序列的總長約為 30 億，那麼兩條人類基因體序列的對齊總個數會是一個超級天文數字。即使利用全世界所有的電腦來做這件事情，我們恐怕都無法在有生之年看到結果，這對電腦而言簡直是「不可能的任務」！幸好，我們可以利用動態規劃 (dynamic programming) 技巧，設計出一個簡單又有效率的演算法來解決兩條序列比對的問題。

資料來源：《科學發展》2005 年 12 月，396 期，36 ~ 41 頁
取自 <https://scitechvista.nat.gov.tw/c/s2Q4.htm>

現在，妳的任務就是協助生物學家進行基因序列比對，搜尋出兩條序列間最長的共同子序列 (subsequence)，方便生物學家進行基因序列的對齊。子序列 (subsequence) 的字母不需要在原來字串裡鄰近出現，只需要保持字母的順序。

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | C | C | G | G | T | G | G | A | C | A | A | T | T | C | A |
| G | G | A | A | A | G | A | G | A | T | A | T | G | C | A | C |

輸入說明

每組測資有兩行，分別代表兩個基因序列 A, B。基因序列由大寫 GACT 構成，每個基因的序列長度 L ($1 \leq L \leq 1000$)。

輸出說明

輸出有兩行。第一行為基因最長共同子序列長度，第二行為此兩條基因的最長共同子序列。

範例輸入

```
ACCGGTGGACAATTCA
GGAAAGAGATATGCAC
```

範例輸出

```
9
GGGAATCA
```



P2. 青蛙變王子 (檔案名稱：P2.CPP)

(動態規劃 ★★★)

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。記得，今年夏天，蛙蛙通過一片具有神奇力量的荷塘，在高高低低的荷葉彈跳挑戰中獲得許多三葉草，也換取不少隱藏物品。經過了 100 天的旅程，蛙蛙終於集滿了所有隱藏物品，來到巫婆居住的冰冷城堡。可是，城堡非常高，蛙蛙必須利用城堡外的石像一一彈跳，累積動能，才能跳上去。

蛙蛙選取的石像高度，只要比上一個石像高，就會增加 1 倍的彈跳動能。現在，我們一起來幫助蛙蛙選擇石像，產生最大動能，順利跳上冰冷城堡，變回帥氣的王子吧!!

輸入說明

第一行輸入的是一個整數 N ($N \leq 30$)，表示接下來有 N 尊石像。

下一行有 N 個整數，依序為每一尊石像的高度 h ($0 < h < 100$)。

輸出說明

輸出有 2 行。

第 1 行為 1 個整數值，代表能讓蛙蛙產生最大動能的石像個數。

第 2 行為 1 個整數值，代表依遞增選取石像的高度差總和。

範例輸入 1

```
5
1 2 3 4 3
```

範例輸出 1

```
4
3
```

範例輸入 2

```
20
10 12 30 14 36 18 20 25 48 32 79 21 49 78 98 22 1 99 97 2
```

範例輸出 2

```
11
89
```



P3. 寶可夢訓練師 (P3.cpp)

(動態規劃 ★★★)

小智夢想成為世界第一的寶可夢訓練師，大木博士給他一個可承重 100 公斤的神奇背包和神奇寶貝球，讓他可以用寶貝球收服寶可夢，裝進神奇背包帶回去訓練。寶可夢的進化值增加量，就是寶可夢訓練師的經驗值，因此，帶走愈有潛力的寶可夢愈好。



小智是個熱心熱血、有著堅強意志的少年，同時也是個頑皮又浮躁的新手訓練師。剛出發旅行的小智缺乏經驗，他常常是見到一隻收服一隻，完全不考慮寶可夢進化強度增加值有多高。這樣，距離他的世界第一訓練師實在很遙遠。大木博士希望你能幫幫小智儘速實現夢想。看到眼前多隻寶可夢出現時，可以在有限背包承重中，聰明選擇該收服與訓練的寶可夢，得到寶可夢訓練師的最大經驗值。

輸入說明

第一行為正整數 N ($0 < N < 30$)，代表寶可夢個數。

接下來 N 行，每行有 3 個整數值 W 、 $CP1$ 、 $CP2$ ，分別代表每隻寶可夢的重量 W 公斤 ($0 < W < 1000$)，進化前的強度值 $CP1$ 與進化後的強度值 $CP2$ 。

輸出說明

輸出 2 個正整數，以 1 個半形空白隔開。分別為收服的寶可夢個數，與寶可夢訓練師的最大經驗值 (訓練師經驗值即寶可夢進化前後強度增加量總和)。

範例輸入

```
5
45 3990 8890
15 8890 11150
39 5680 18300
47 10690 15120
50 15120 21080
```

範例輸出

```
3 19780
```



P4. N 度分隔理論 (P4.cpp)

(圖論★★★)

所謂的「六度分隔理論」(Six Degrees of Separation) 為哈佛大學心理系 Stanley Milgram 教授於 1967 年所提出的一套理論，他認為每個人都如同是一張蜘蛛網上的一個結點，每個結點都在其他結點連在一起，進而構成世界人脈這張大網。該理論認為，你和全世界任何一個人的間隔關係都不會超過六個人；換言之，最多只要透過六個人，你就可以連結到全世界任何一個人。

5

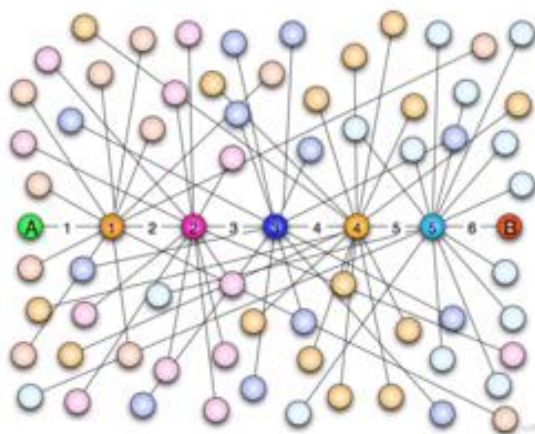


Figure 1 六度分隔理論

(資料來源：https://en.wikipedia.org/wiki/Six_degrees_of_separation)

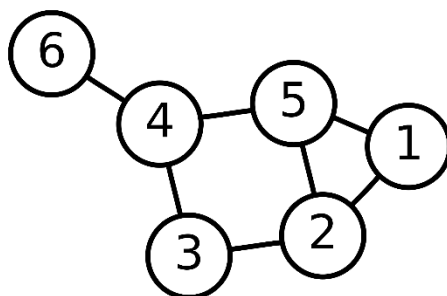
當他在 1967 年提出這套理論時，全球化網路與社群網站尚未成形，所以這個理論難以印證。沉寂近三十年後，這套理論因為同名電影而突然爆紅，從學術紙堆中被翻出，成為大多數資訊專家都琅琅上口的一套說詞，今天 Facebook 之類的 SNS 網路社群就是六度分隔理論的最好證明，朋友的朋友剛好又是親人的弟弟的朋友，女友的妹妹的朋友的表弟就是姊姊的家人，世界如此之小。

不過短短數年不到，這個擁有四十年歷史的理論已經過時，因為根據最近的一個研究，現在是「四度分隔理論」了。現在你與全世界任何一個人的聯繫，只需要四個人就可以達成。facebook 的數據資料庫和流量訊息、地域關係，已經證實全世界每一個人之間的聯繫間隔環已經縮減到 3.57 人。在一些網路高度化的國家，這個數字會降低；如果是連結本國或同語文的「陌生人」，這個數字還會更下降，看來 Benedict Anderson 所謂的「想像共同體的固有邊界，正在逐漸消融冰解中，我們可以預測，隨著全球網路化與資訊的普及，這個數字還會不斷下降，「四海之內皆兄弟」已將不再是俗諺。

文本資料來源：六度分隔理論已經過時，現在是最熟悉的陌生人的時代



現在，我們定義 **N 度分隔** 的 N 為「分離度平均值」。



例如：上圖為 6 個人的人際網路關係圖，「分離度平均值」即是把每個人和他人的最短分離度相加之後，除以所有關係數所得的結果。圖中任意 2 人的最短分離度如下表，每個人和他人的最短分離度相加總和為 50，其中存在 $6 \times 5 = 30$ 個關係數，分離度平均值為 1.67。我們稱此 6 個人的人際網路存在 **1.67 度分隔**。

| 任 2 人的分離度 | | | | |
|------------|------------|------------|------------|------------|
| (1, 2) = 1 | (1, 3) = 2 | (1, 4) = 2 | (1, 5) = 1 | (1, 6) = 3 |
| (2, 1) = 1 | (2, 3) = 1 | (2, 4) = 2 | (2, 5) = 1 | (2, 6) = 3 |
| (3, 1) = 2 | (3, 2) = 1 | (3, 4) = 1 | (3, 5) = 2 | (3, 6) = 2 |
| (4, 1) = 2 | (4, 2) = 2 | (4, 3) = 1 | (4, 5) = 1 | (4, 6) = 1 |
| (5, 1) = 1 | (5, 2) = 1 | (5, 3) = 2 | (5, 4) = 1 | (5, 6) = 2 |
| (6, 1) = 3 | (6, 2) = 3 | (6, 3) = 2 | (6, 4) = 1 | (6, 5) = 2 |

現在，給妳 n 個人的人際關係網路，請妳試算在這個關係網路中，人與人之間的為幾度分隔？（本題假設每一個人都至少有 1 個朋友）

輸入說明

第一行有一個整數 n ($1 \leq n \leq 10000$) 代表人際關係網路圖的節點數量。

接下來有 n 行，每行有 n 個整數 R (R 為 0 或 1)，代表每個人與他人的關係。

輸出說明

輸出有 1 行。代表此人際網路關係的分離度平均值 (輸出至小數點後 2 位)。

範例輸入

```
6
0 1 0 0 1 0
1 0 1 0 1 0
0 1 0 1 0 0
0 0 1 0 1 1
1 1 0 1 0 0
0 0 0 1 0 0
```

範例輸出

```
1.67
```



P5. 資料壓縮技術 (p5.cpp)

(字串★★)

隨著數位科技的進步與網際網路的蓬勃發展，包含影像、聲音與視訊的各種多媒體應用技術不斷推陳出新。由於多媒體資料的龐大，資料壓縮就顯得重要且無法避免。資料壓縮不僅減少儲存空間，傳輸時間也是大大縮短。

變動長度編碼法 (RLE, Run Length Encoding) 是常常被使用的資料壓縮技術，原理是使用變動長度編碼來取代連續重複出現的原始資料。舉例來說，一組資料串 "AAAABBBCCDEEEE"，由 4 個 A、3 個 B、2 個 C、1 個 D、4 個 E 組成，經過變動長度編碼法可將資料壓縮為 4A3B2C1D4E (資料長度由 14 個單位轉成 10 個單位)。其優點在於將重複性高的資料量壓縮成小單位；然而，其缺點在於——若該資料出現頻率不高，可能導致壓縮結果資料量比原始資料大，例如：原始資料 "ABCDE"，壓縮結果為 "1A1B1C1D1E" (資料長度由 5 個單位轉成 10 個單位)。

在本問題中，我們感興趣的是請妳將編碼過的資料字串進行解碼，還原為未壓縮前的訊息。

輸入說明

第一行輸入是整數 T ($T < 50$)，表示編碼字串的數量。

接下來 T 行，每行都是由 RLE 編碼組成的字串，字串中僅包含數字 [0-9] 和字母 [A-Z]。每個字母後面都會跟一個或多個數字。詳如範例輸入。

輸出說明

輸出有 T 行。

每行輸出對應解碼後的字串。

你可以假設解碼後的字串長度不會超過 200，並且只包含大寫字母。

範例輸入

```
3
A2B4D1A2
A12
A1B1C1D1
```

範例輸出

```
AABBBBDAA
AAAAAAAAAAAA
ABCD
```

