

大學程式設計先修檢測

2017.10.28

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案附檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

邏輯運算子 (Logic Operators)

問題描述

小蘇最近在學三種邏輯運算子 AND、OR 和 XOR。這三種運算子都是二元運算子，也就是說在運算時需要兩個運算元，例如 $a \text{ AND } b$ 。對於整數 a 與 b ，以下三個二元運算子的運算結果定義如下列三個表格：

a AND b		
	b 為 0	b 不為 0
a 為 0	0	0
a 不為 0	0	1

a OR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	1

a XOR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	0

舉例來說：

- (1) $0 \text{ AND } 0$ 的結果為 0， $0 \text{ OR } 0$ 以及 $0 \text{ XOR } 0$ 的結果也為 0。
- (2) $0 \text{ AND } 3$ 的結果為 0， $0 \text{ OR } 3$ 以及 $0 \text{ XOR } 3$ 的結果則為 1。
- (3) $4 \text{ AND } 9$ 的結果為 1， $4 \text{ OR } 9$ 的結果也為 1，但 $4 \text{ XOR } 9$ 的結果為 0。

請撰寫一個程式，讀入 a 、 b 以及邏輯運算的結果，輸出可能的邏輯運算為何。

輸入格式

輸入只有一行，共三個整數值，整數間以一個空白隔開。第一個整數代表 a ，第二個整數代表 b ，這兩數均為非負的整數。第三個整數代表邏輯運算的結果，只會是 0 或 1。

輸出格式

輸出可能得到指定結果的運算，若有多個，輸出順序為 AND、OR、XOR，每個可能的運算單獨輸出一行，每行結尾皆有換行。若不可能得到指定結果，輸出 IMPOSSIBLE。（注意輸出時所有英文字母均為大寫字母。）

範例一：輸入

0 0 0

範例一：正確輸出

AND

OR

XOR

範例二：輸入

1 1 1

範例二：正確輸出

AND

OR

範例三：輸入

3 0 1

範例三：正確輸出

OR

XOR

範例四：輸入

0 0 1

範例四：正確輸出

IMPOSSIBLE

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 80 分， a 和 b 的值只會是 0 或 1。

第 2 子題組 20 分， $0 \leq a, b < 10,000$ 。

大學程式設計先修檢測

2017.10.28

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案附檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

交錯字串 (Alternating Strings)

問題描述

一個字串如果全由大寫英文字母組成，我們稱為大寫字串；如果全由小寫字母組成則稱為小寫字串。字串的長度是它所包含字母的個數，在本題中，字串均由大小寫英文字母組成。假設 k 是一個自然數，一個字串被稱為「 k -交錯字串」，如果它是由長度為 k 的大寫字串與長度為 k 的小寫字串交錯串接組成。

舉例來說，「StRiNg」是一個 1-交錯字串，因為它是一個大寫一個小寫交替出現；而「heLLow」是一個 2-交錯字串，因為它是兩個小寫接兩個大寫再接兩個小寫。但不管 k 是多少，「aBBaaa」、「BaBaBB」、「aaaAAbbCCCC」都不是 k -交錯字串。

本題的目標是對於給定 k 值，在一個輸入字串找出最長一段連續子字串滿足 k -交錯字串的要求。例如 $k=2$ 且輸入「aBBaaa」，最長的 k -交錯字串是「BBaa」，長度為 4。又如 $k=1$ 且輸入「BaBaBB」，最長的 k -交錯字串是「BaBaB」，長度為 5。

請注意，滿足條件的子字串可能只包含一段小寫或大寫字母而無交替，如範例二。此外，也可能不存在滿足條件的子字串，如範例四。

輸入格式

輸入的第一行是 k ，第二行是輸入字串，字串長度至少為 1，只由大小寫英文字母組成(A~Z, a~z)並且沒有空白。

輸出格式

輸出輸入字串中滿足 k -交錯字串的要求的最長一段連續子字串的長度，以換行結尾。

範例一：輸入

1
aBBdaaa

範例一：正確輸出

2

範例三：輸入

2
aaFAXbbCDCCC

範例三：正確輸出

8

範例二：輸入

3
DDaasAAbbCC

範例二：正確輸出

3

範例四：輸入

3
DDaaAAbbCC

範例四：正確輸出

0

評分說明：

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分，字串長度不超過 20 且 $k=1$ 。

第 2 子題組 30 分，字串長度不超過 100 且 $k \leq 2$ 。

第 3 子題組 50 分，字串長度不超過 100,000 且無其他限制。

提示：根據定義，要找的答案是大寫片段與小寫片段交錯串接而成。本題有多種解法的思考方式，其中一種是從左往右掃描輸入字串，我們需要紀錄的狀態包含：目前是在小寫子字串中還是大寫子字串中，以及在目前大(小)寫子字串的第幾個位置。根據下一個字母的

大小寫，我們需要更新狀態並且記錄以此位置為結尾的最長交替字串長度。

另外一種思考是先掃描一遍字串，找出每一個連續大(小)寫片段的長度並將其記錄在一個陣列，然後針對這個陣列來找出答案。

大學程式設計先修檢測

2017.10.28

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案附檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

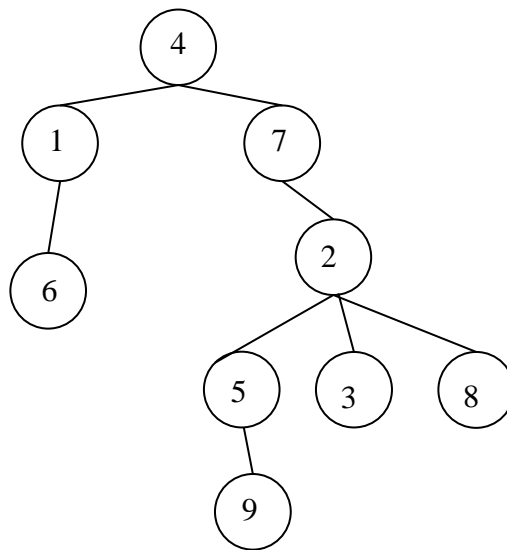
2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

樹狀圖分析 (Tree Analyses)

問題描述

本題是關於有根樹(rooted tree)。在一棵 n 個節點的有根樹中，每個節點都是以 $1 \sim n$ 的不同數字來編號，描述一棵有根樹必須定義節點與節點之間的親子關係。一棵有根樹恰有一個節點沒有父節點(parent)，此節點被稱為根節點(root)，除了根節點以外的每一個節點都恰有一個父節點，而每個節點被稱為是它父節點的子節點(child)，有些節點沒有子節點，這些節點稱為葉節點(leaf)。在當有根樹只有一個節點時，這個節點既是根節點同時也是葉節點。

在圖形表示上，我們將父節點畫在子節點之上，中間畫一條邊(edge)連結。例如，圖一中表示的是一棵 9 個節點的有根樹，其中，節點 1 為節點 6 的父節點，而節點 6 為節點 1 的子節點；又 5、3 與 8 都是 2 的子節點。節點 4 沒有父節點，所以節點 4 是根節點；而 6、9、3 與 8 都是葉節點。



圖一

樹狀圖中的兩個節點 u 和 v 之間的距離 $d(u,v)$ 定義為兩節點之間邊的數量。如圖一中， $d(7, 5) = 2$ ，而 $d(1, 2) = 3$ 。對於樹狀圖中的節點 v ，我們以 $h(v)$ 代表節點 v 的高度，其定義是節點 v 和節點 v 下面最遠的葉節點之間的距離，而葉節點的高度定義為 0。如圖一中，節點 6 的高度為 0，節點 2 的高度為 2，而節點 4 的高度為 4。此外，我們定義 $H(T)$ 為 T 中所有節點的高度總和，也就是說 $H(T) = \sum_{v \in T} h(v)$ 。給定一個樹狀圖 T ，請找出 T 的根節點以及高度總和 $H(T)$ 。

輸入格式

第一行有一個正整數 n 代表樹狀圖的節點個數，節點的編號為 1 到 n 。接下來有 n 行，第 i 行的第一個數字 k 代表節點 i 有 k 個子節點，第 i 行接下來的 k 個數字就是這些子節點的編號。每一行的相鄰數字間以空白隔開。

輸出格式

輸出兩行各含一個整數，第一行是根節點的編號，第二行是 $H(T)$ 。

範例一：輸入

```
7
0
2 6 7
2 1 4
0
2 3 2
0
0
```

範例一：正確輸出

```
5
4
```

範例二：輸入

```
9
1 6
3 5 3 8
0
2 1 7
1 9
0
1 2
0
0
```

範例二：正確輸出

```
4
11
```

評分說明：

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。測資範圍如下，其中 k 是每個節點的子節點數量上限：

第 1 子題組 10 分， $1 \leq n \leq 4, k \leq 3$ ，除了根節點之外都是葉節點。

第 2 子題組 30 分， $1 \leq n \leq 1,000, k \leq 3$ 。

第 3 子題組 30 分， $1 \leq n \leq 100,000, k \leq 3$ 。

第 4 子題組 30 分， $1 \leq n \leq 100,000, k$ 無限制。

提示：輸入的資料是給每個節點的子節點有哪些或沒有子節點，因此，可以根據定義找出根節點。關於節點高度的計算，我們根據定義可以找出以下遞迴關係式：(1)葉節點的高度為 0；(2)如果 v 不是葉節點，則 v 的高度是它所有子節點的最大高度加一。也就是說，假設 v 的子節點有 a, b 與 c ，則 $h(v) = \max\{h(a), h(b), h(c)\} + 1$ 。以遞迴方式可以計算出所有節點的高度。

大學程式設計先修檢測

2017.10.28

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案附檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

物品堆疊(Stacking)

問題描述

某個自動化系統中有一個存取物品的子系統，該系統是將 N 個物品堆在一個垂直的貨架上，每個物品各佔一層。系統運作的方式如下：每次只會取用一個物品，取用時必須先將在其上方的物品貨架升高，取用後必須將該物品放回，然後將剛才升起的貨架降回原始位置，之後才會進行下一個物品的取用。

每一次升高某些物品所需要消耗的能量是以這些物品的總重來計算，在此我們忽略貨架的重量以及其他可能的消耗。現在有 N 個物品，第 i 個物品的重量是 $w(i)$ 而需要取用的次數為 $f(i)$ ，我們需要決定如何擺放這些物品的順序來讓消耗的能量越小越好。舉例來說，有兩個物品 $w(1)=1$ 、 $w(2)=2$ 、 $f(1)=3$ 、 $f(2)=4$ ，也就是說物品 1 的重量是 1 需取用 3 次，物品 2 的重量是 2 需取用 4 次。我們有兩個可能的擺放順序(由上而下)：

- (1,2)，也就是物品 1 放在上方，2 在下方。那麼，取用 1 的時候不需要能量，而每次取用 2 的能量消耗是 $w(1)=1$ ，因為 2 需取用 $f(2)=4$ 次，所以消耗能量數為 $w(1)*f(2)=4$ 。
- (2,1)，也就是物品 2 放在 1 的上方。那麼，取用 2 的時候不需要能量，而每次取用 1 的能量消耗是 $w(2)=2$ ，因為 1 需取用 $f(1)=3$ 次，所以消耗能量數 $=w(2)*f(1)=6$ 。

在所有可能的兩種擺放順序中，最少的能量是 4，所以答案是 4。再舉一例，若有三物品而 $w(1)=3$ 、 $w(2)=4$ 、 $w(3)=5$ 、 $f(1)=1$ 、 $f(2)=2$ 、 $f(3)=3$ 。假設由上而下以 (3,2,1) 的順序，此時能量計算方式如下：取用物品 3 不需要能量，取用物品 2 消耗 $w(3)*f(2)=10$ ，取用物品 1 消耗 $(w(3)+w(2))*f(1)=9$ ，總計能量為 19。如果以 (1,2,3) 的順序，則消耗能量為 $3*2+(3+4)*3=27$ 。事實上，我們一共有 $3!=6$ 種可能的擺放順序，其中順序 (3,2,1) 可以得到最小消耗能量 19。

輸入格式

輸入的第一行是物品件數 N ，第二行有 N 個正整數，依序是各物品的重量 $w(1)$ 、 $w(2)$ 、...、 $w(N)$ ，重量皆不超過 1000 且以一個空白間隔。第三行有 N 個正整數，依序是各物品的取用次數 $f(1)$ 、 $f(2)$ 、...、 $f(N)$ ，次數皆為 1000 以內的正整數，以一個空白間隔。

輸出格式

輸出最小能量消耗值，以換行結尾。所求答案不會超過 63 個位元所能表示的正整數。

範例一(第 1、3 子題)：輸入

```
2
20 10
1 1
```

範例一：正確輸出

```
10
```

範例二(第 2、4 子題)：輸入

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

範例二：正確輸出

```
19
```

評分說明：輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 10 分， $N=2$ ，且取用次數 $f(1)=f(2)=1$ 。

第 2 子題組 20 分， $N=3$ 。

第 3 子題組 45 分， $N \leq 1,000$ ，且每一個物品 i 的取用次數 $f(i)=1$ 。

第 4 子題組 25 分， $N \leq 100,000$ 。