

## 上機習題 #2 (建造 C++ 的集合類別)

題目：以 C++ 製作一個「集合」類別 (set class)，使其能進行「聯集」、「交集」、「差集」、「包含」、「屬於」等五種運算。

說明：

Pascal 本身有提供集合運算，但是 C 語言並沒有，故請你製作集合類別，以利進行集合運算。集合運算符號與 Pascal 相同。

假設 A 與 B 均為集合，則以下列符號表示各種運算：

A 與 B 聯集 (union) 記做  $A + B$

A 與 B 交集 (intersection) 記做  $A * B$

A 與 B 差集 (difference) 記做  $A - B$

A 包含 (contain) B 記做  $A \supseteq B$

x 屬於 (belong to) A 記做  $x \in A$

為了簡化設計，集合元素僅需為字元(char)型態。集合內部資料結構可用字元陣列(char array)來表示，由於廣義的字元(ASCII code)有 256 個字元，故可使用 長度為 256 的陣列來表示一個集合。

本習題必須使用「運算子重載」(operator overloading, 即必須重新定義  $+$ 、 $-$ 、 $*$ 、 $\supseteq$  的運算法則)。假設你的集合類別稱為 TSet，你的集合必須能直接透過運算子進行以下運算：

TSet A, B, C, D; //宣告集合

$C = A + B$ ; //把 A 和 B 聯集的結果放入 C

$D = A * B$ ; //把 A 和 B 交集的結果放入 D

輸入格式：

有好幾組測試資料，第一列為測試資料組數 N， $1 \leq N \leq 20$ 。每一組測試資料共有三列，每一組的第一列與第二列(每列至多為 80 個字元)，分別為集合 A 與集合 B 之元素；第三列為一個字元 x，欲測試 x 是否在集合 A、是否在集合 B 之內。請注意：(1) 輸入的字元可能包含空白字元或其他符號，但不包含換行符號。(2) 一整列屬於同一個集合。(3) 輸入的字元沒有排序。(4) 可能有重複的字元，重複的字元在一個集合只能算一次。

輸出格式：

對於每一組測試資料，首先印出測試組編號 Test Case #，接著印出集合 A 與集合 B 之內容(元素必須依照 ASCII 編碼，由小至大印出。若有重複，僅能印出一次)，然後再印出上述五種運算結果。兩組測試資料之間以空白列隔開。

輸入範例：

```
2
abcdef
chfeechi
h
3abf4ec
43
g
```

輸出範例：

Test Case 1:

A: {abcdef}

B: {cefhi}

A+B: {abcdefhi}

A\*B: {cef}

A-B: {abd}

B-A: {hi}

A does not contain B

B does not contain A

'h' is not in A

'h' is in B

Test Case 2:

A: {34abcef}

B: {34}

A+B: {34abcef}

A\*B: {34}

A-B: {abcef}

B-A: {}

A contains B

B does not contain A

'g' is not in A

'g' is not in B

特別注意：

助教在測試時，保證所輸入的資料格式完全正確，故無需檢驗資料之正確性。程式碼的部份，請務必利用重載運算子來實作，否則不予計分。