Assignment #2 (Establishment of the set class by C++)

Goal：Use C++ to implement a set class, which provides union, intersection, difference, containment and belonging, as basic operations.

Guide：

The Pascal language provides the operation for set classes, while the C language does not. Therefore, you are asked to implement a set class with C, which facilitates the operations on sets. The symbols used in the operators of your set class are the same as those in Pascal.

Assume A and B are both sets. The following symbols are used to define each operation:

The union of A and B is represented as “A + B”
The intersection of A and B is represented as “A * B”
The difference of A and B is represented as “A – B”
”A contains B” is represented as “A >= B”
“x belongs to A” is represented as “x in A”

For simplicity, each element in this assignment is a single character. That is, you can use long bit string to store all elements in a set. Because the ASCII code has 256 distinct values, you can use an array of size 256 to represent a set.

In this assignment, you have to overload the operators (define what your operators are and what they can do). Suppose that you have implemented a set class named TSet, then other programmers should be able to take advantage of your implementation. For example, he should be able to write the following codes.

```
TSet A, B, C, D;   //declaration for the class TSet
C = A+B;   //Store the union of A and B into C
D = A*B;   //Store the intersection of A and B into D
```

Input format：

There are several test cases. The number N in the first means the number of test cases, where 1≤N≤20. There are three lines for each test case. In each test case, the first line and the second line (each line has up to 80 characters) are the elements of sets A and B, respectively. The third line contains a single character x, which is to be tested in the set A or set B. Note: (1) Each input character may be a blank characters or other symbols, but not a newline character. (2) The characters in one line belong to the same set. (3) The input characters are not ordered. (4) There may be some duplicated characters, but the same duplicated characters in a set can be counted only once.

Output format：

For each test case, first print the case number “Test Case #”, and then print the contents of sets A and B (The elements must be sorted with the ASCII code, from small to large. If there are duplicated characters, print it only once.), and finally print out the results of the five operations. Print a blank line between two test cases.

Sample input:

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

Sample output:

```
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

Notice:

You can assume that the input from TA will always be valid. In other words, you do not have to check if the inputs are valid or not. You have to implement your set class by overriding operators. Otherwise, your program will not be accepted.