The Northeast Regional Competition
of the
2006-2007 ACM International Collegiate Programming Contest
Western New England College Site
October 14, 2006

Problem #5: Insiders

There are several specification models, such as context-free grammars, regular sets and Turing machines, in formal languages and automata theory. In this problem you are to write a program to handle regular sets.

Let Σ be a finite set of alphabets. **Regular sets** over Σ are defined as follows:

(1) $\varnothing$, $\{\lambda\}$ and $\{a\}$, for every a in Σ, are regular sets over Σ. (λ is the null string.)

(2) If X and Y are regular sets over Σ, then $X \cup Y$, XY and X^* are also regular sets over Σ.

Recall that

$X \cup Y = \{u \mid u \in X \text{ or } u \in Y\}$ is the **union** of two sets

$XY = \{uv \mid u \in X \text{ and } v \in Y\}$ is the **concatenation** of two sets

X^* = the set of all strings over X, including the null string λ

In this problem we assume that $\Sigma = \{a, b, c, d, e\}$. We will use an uppercase letter to denote a singleton set containing the corresponding lowercase letter, such as A = {a}, B = {b}, and so on. Also, we will use + to denote a union, and X* to denote X^*. Here are some regular sets over Σ:

ABB = {a}{b}{b} = {abb}

(ABB)* = $(\{a\}\{b\}\{b\})^*$ = $\{abb\}^*$ = {abb, abbabb, abbabbabb, ...}

(A+B)* = $(\{a\}\cup\{b\})^*$ = $\{a, b\}^*$ = set of all strings over {a, b}

AB* = $\{a\}\{b\}^*$ = $\{a\}\{\lambda, b, bb, bbb, ...\}$ = {a, ab, abb, abbb, ...}

$A(A+B+C+D+E)^*BB$ = set of all strings that begin with a and end with bb

Note that among the three operators, * has the highest precedence, followed by concatenation, and union has the lowest precedence. Parentheses are used to override these precedence rules. So AB* is A(B*), not (AB)*.

Your program is to determine whether any given string is in any given regular set. The first line of the input is an integer indicating the number of regular sets your program has to process. Each regular set is followed by one or more strings. Each string is on a separate line, and these unknown number of strings for each regular set will be terminated by a sentinel line containing a single character '@'. You may assume that each regular set contains a maximum of 30 characters, including uppercase letters, '(', ')', '*', and '+'. Also, each string contains a maximum of 50 lowercase letters. Your program should output each regular set followed by strings in that set. If none of the input strings is in the regular set, your program should print the message **"no string found"**. Also, you should **use a blank line to separate two regular sets** in your output.

Sample Input

```
4
AB
acbbd
ab
@
B(A+B)*C
baabac
bc
bbcac
@
((ABC)+A)B*
ac
aa
@
AB(C(A+B)*+E)*
ab
aaebc
abcabbbaec
abeecbababbab
@
```

Sample Output

```
AB
ab

B(A+B)*C
baabac
bc

((ABC)+A)B*
no string found

AB(C(A+B)*+E)*
ab
abcabbbaec
abeecbababbab
```