Problem I

A Grammarian to the Rescue

Time Limit=1 second

In year 9999, human race becomes extinct and an alien life form so-called *Borg*[1] comes to Earth to investigate the historic rationale behind this bizarre incident. Dr. Jumbo Jookiba leads the research team to locate remnants of scientific records elaborating antecedents of the happening. Using Borg algorithm[2], the team locates two major findings for the causes in Bandung, West Java, specifically at Maranatha Christian University (MCU).

The first momentous discovery is situated at Faculty of Letters of MCU. The artifact contains a grammar. Dr. Jookiba hyphotesize that in order to understand why the preposterous occurrence took place, the team needs to comprehend and decipher the grammar. A sample of the grammar (Jurafsky and Martin, 2008) is described in Table 1.

Table 1: A Grammar

Grammar in Chomsky normal form (CNF)	
$S \to NP\ \ VP$	$Nom \to book \mid flight \mid meal \mid money$
$S \to X1\ \ VP$	$Nom \to Nom\ Noun$
$X1 \to Aux\ NP$	$Nom \to Nom\ PP$
$S \to book \mid include \mid prefer$	$VP \to book \mid include \mid prefer$
$S \to Verb\ \ NP$	$VP \to Verb\ NP$
$S \to X2\ \ PP$	$VP \to X2\ PP$
$S \to Verb\ \ PP$	$X2 \to Verb\ NP$
$S \to VP\ \ PP$	$VP \to Verb\ PP$
$NP \to I \mid she \mid me$	$VP \to VP\ PP$
$NP \to TWA \mid Houston$	$PP \to Prep\ NP$
$NP \to Det\ \ Nom$	

After studying with stupendous effort, the team succeeds in discovering the meaning of each terms and understanding the concept of Chomsky normal form (CNF). Grammars in CNF are restricted to rules of the form $A \to B\ C$ or $A \to w$. That is, the right-hand side of each rule must expand either to two non-terminals or to a single terminal (Jurafsky and Martin, 2008).

A term is called a terminal if it cannot be expanded into other terms, for example, $book$ and $flight$ are terminal terms. Conversely, a term is a non-terminal if it can be elaborated into other terms, for instance, S is a non-terminal since it can be decomposed into VP and PP.

The expansion of abbreviations in Table 1 is given as follows:

[1] https://en.wikipedia.org/wiki/Borg_(Star_Trek)
[2] http://memory-alpha.wikia.com/wiki/Borg_algorithm

S	=	sentence, for example, *Mom made chicken for dinner*.
NP	=	noun phrase, for example, *a new fishing pole*.
VP	=	verb phrase, for example, *sings beautifully*.
$S \rightarrow NP\ \ VP$	=	a sentence can be decomposed into NP and VP.
PP	=	prepositional phrase, for example, *with him, in black*.
Det	=	determiner, for example, *a, an, the*.
Aux	=	auxiliary, for example, *does, do*.
Nom	=	nominal; used as a noun.

Having discovered the meaning of each terms, Dr. Jookiba postulates that there is a *mystical* pattern that connects all these terms. Such pattern is captured by a diagram defined as a *parse tree*. For example, a sentence *Book that flight* has a parse tree depicted in Figure 1.

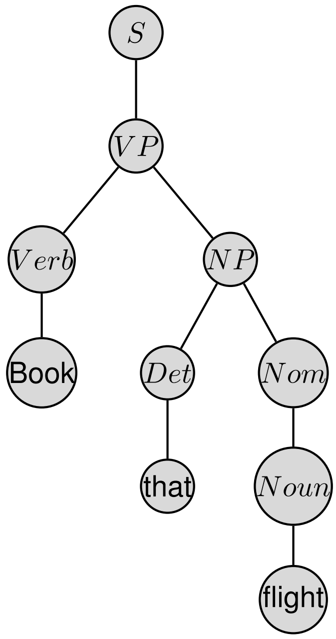

Figure 1: The parse tree for the sentence *Book that flight* according to Table 1.

The last but not least crucial unearthing is located at Faculty of Information Technology of MCU. The team finds a remains of CKY parse algorithm delineated in Algorithm 1. Unfortunately, the Algorithm 1 is a *recognizer*, not a *parser*. A recognizer just recognizes whether or not a sentence can be parsed; on the other hand, a parser parses a sentence into a parse tree. In order to make a parser, the recognizer should also have to find an S in cell $[0, N]$ of the $table$ in Algorithm 1.

The creator of this problem forgets to mention that there is a human in Dr. Jookiba's research team. That human is you and you are ***the grammarian***. Therefore, you have to help Dr. Jookiba. The future of understanding the catastrophic event is in your hand. Good luck!

Algorithm 1 The CKY algorithm

function CKY-PARSE($words, grammar$)
    ▷ **Output**: *A* CKY *table, an* $n \times n$ *array.*
    for $j \leftarrow 1$ **to** LENGTH($words$) **do**
        $table[j-1, j] \leftarrow \{A \mid A \rightarrow words[j] \in grammar\}$
        for $i \leftarrow j-2$ **downto** 0 **do**
            for $k \leftarrow i+1$ **to** $j-1$ **do**
                $table[i, j] \leftarrow table[i, j] \cup \{A|A \rightarrow BC \in grammar,$
                    $B \in table[i, k],$
                    $C \in table[k, j]\}$
            end for
        end for
    end for
    return $table$
end function

Input

The input will consist of information for a grammar and a sentence and the details are as follows:

a) a line containing 1 integer ($0 < n \leq 50$): n is the number of rules.
b) n lines where each line is a rule. Each must follow CNF. Each rule is a string up to 50 characters long, terminated by the end of line. All strings are case sensitive.
c) A line describes a sentence that needs parsing. The sentence is is a string up to 50 characters long, terminated by the end of line.

Output

The output gives a list of possible parse(s).

Sample Input

```
16
S -> NP VP
S -> X1 VP
X1 -> Aux NP
S -> 'book'
PP -> P NP
NP -> Det N
NP -> Det N PP
NP -> 'I'
VP -> V NP
VP -> VP PP
Det -> 'an'
Det -> 'my'
N -> 'elephant'
N -> 'pajamas'
V -> 'shot'
P -> 'in'
I shot an elephant in my pajamas
```

Sample Output

```
Possible parse(s):
[S [NP 'I'][VP [V 'shot'][NP [NP0 [Det 'an'][N 'elephant']][PP [P
'in'][NP [Det 'my'][N 'pajamas']]]]]]
[S [NP 'I'][VP [VP [V 'shot'][NP [Det 'an'][N 'elephant']]][PP [P
'in'][NP [Det 'my'][N 'pajamas']]]]]
```

Another Sample Input

```
12
S -> NP VP
VP -> VP PP
VP -> Verb NP
PP -> P NP
NP -> Det N
NP -> 'Mary'
Verb -> 'rode'
Det -> 'her'
Det -> 'the'
N -> 'bike'
P -> 'to'
N -> 'store'
Mary rode her bike to the store
```

Another Sample Output

```
Possible parse(s):
[S [NP 'Mary'][VP [VP [Verb 'rode'][NP [Det 'her'][N 'bike']]][PP
[P 'to'][NP [Det 'the'][N 'store']]]]]
```

References

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing, 2nd Edition*. Prentice Hall.