

# 2023 Compiler Boss Attack 1

Please write a program to read a program source from **stdin** following the token definition and grammar rule at right.

If **Yes**, print each **the string of token and token's type** separated by a **whitespace " "** and end with a **newline**.

If **No**, print only **"invalid input"** with a **newline**. (*don't output any token!*)

The purpose of this boss attack is to test your understanding in writing a **recursive descent parser**. So, it is forbidden to use lex, yacc, or any other parser generator tools. Also, **"#include<regex>" is NOT allowed**. **Please write a program using the concept of recursive descent parser in C/C++; otherwise, your score will be zero.**

Terminal	Regular Expression
ID	<code>^[A-Z]\d{9}</code>
NATIONNUM	<code>\d{3}</code>
PHONENUM	<code>9[0-9]{8}</code>
YEAR	<code>[12]\d{3}</code>
MONTH	<code>(JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC)</code>
DAY	<code>[1-9]  [12]\d 3[01]</code>
PLUS	<code>\+</code>
DOT	<code>\.</code>
LBR	<code>\(</code>
RBR	<code>\)</code>
SLASH	<code>\/</code>

Notice:

1. There is **NO** need to distinguish between leap years and non-leap years. You can determine it using Regular Expressions. For example, both "FEB 31" and "FEB 29" are considered valid.
2. The test data will **NOT** contain whitespace characters.
3. Each test case will **only have one line**.

1. program → stmt
2. stmt → ID
3. stmt → phone
4. stmt → date
5. phone → PLUS LBR NATIONNUM RBR PHONENUM
6. date → MONTH DOT DAY DOT YEAR
7. date → YEAR SLASH MONTH SLASH DAY

<b>Sample input:</b> JUN.15.2013 <b>Sample output:</b> JUN MONTH . DOT 15 DAY . DOT 2013 YEAR	<b>Sample input:</b> +(886)912123456 <b>Sample output:</b> + PLUS ( LBR 886 NATIONNUM ) RBR 912123456 PHONENUM
<b>Sample input:</b> JUN/15 <b>Sample output:</b> Invalid input	<b>Sample input:</b> C5585489666 <b>Sample output:</b> Invalid input