

B – Parser

Description

Please write a program to read a program source from stdin following the token definition and grammar rule below. Your program has to check the source whether it follows the token and grammar rules or not.

IMPORTANT:

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. We only prepare the C/C++ compiler behind DomJudge.

| Terminal | Regular Expression |
|------------|--------------------|
| num | [0-9] |
| funcName | [a-z] |
| className | [A-Z] |
| leftParen | (|
| rightParen |) |

| Productions |
|---|
| Proc \rightarrow Stmt |
| Stmt \rightarrow Class Func |
| Class \rightarrow className leftParen rightParen Args |
| Func \rightarrow leftParen funcName Args rightParen |
| Args \rightarrow Arg Args λ |
| Arg \rightarrow num Func |

Input Format

The input is a character sequence separated by space and has a newline at its end. There won't be any invalid token in the input stream.

Output Format

You have to check if the character sequence follows the grammar or not.

If the input sequence is **valid**, print each **token's type** and the **content of the token** separated by a **whitespace " "** and end with a **newline**.

If the input sequence is invalid, print only **"Invalid input"** with a **newline**. (don't output any token!)

| | |
|--|---|
| <div>Sample Input</div> <div>(f 1)</div> <div>Sample Output</div> <div>leftParen (funcName f num 1 rightParen)</div> | <div>Sample Input</div> <div>C () 2</div> <div>Sample Output</div> <div>className C leftParen (rightParen) num 2</div> |
| <div>Sample Input</div> <div>(b 2 (a) 0)</div> <div>Sample Output</div> <div>leftParen (funcName b num 2 leftParen (funcName a rightParen) num 0 rightParen)</div> | <div>Sample Input</div> <div>C () (b 1) 3</div> <div>Sample Output</div> <div>className C leftParen (rightParen) leftParen (funcName b num 1 rightParen) num 3</div> |
| <div>Sample Input</div> <div>(f ())</div> <div>Sample Output</div> <div>Invalid input</div> | <div>Sample Input</div> <div>c () (f 3)</div> <div>Sample Output</div> <div>Invalid input</div> |