Christian John L. Legaspi REVISED OUTPUT
CS3B

CFG_Example.cpp output:

```
hyuse@Hyuse:~/Desktop/Automata/Automata Simulations$ make run_cfg<test_input.txt
./Context-Free\ Grammar/CFG_Example
==========================================
   CFG PARSER: Tuples, Lists & Strings
==========================================

FORMAL GRAMMAR DEFINITION G = (V, Sigma, R, S):
  V (Variables): { S, E, L, Q }
  Sigma (Terminals): { a-z, 0-9, " , (, ), [, ], {, }, , }
  R (Production Rules):
    1. S -> E S | epsilon
    2. E -> char | Q | ( L )
    3. Q -> " Content "
    4. L -> E , L | E

------------------------------------------
INSTRUCTIONS:
  - Enter strings to test.
  - Valid:   ("hello", b, [c])
  - Invalid: ("hello, b)   (Missing quote)
------------------------------------------

How many test cases?
Input: a
Result: [ ACCEPTED ]

Input: "hello"
Result: [ ACCEPTED ]

Input: print("Hello World!")
Result: [ ACCEPTED ]

Input: ()
Result: [ ACCEPTED ]

Input: (a)
Result: [ ACCEPTED ]

Input: (a,b)
Result: [ ACCEPTED ]

Input: [a, "b", (c)]
Result: [ ACCEPTED ]

Input: a b "c"
Result: [ ACCEPTED ]

Input: ((a))
Result: [ ACCEPTED ]

Input: (a, (b, c), [d])
Result: [ ACCEPTED ]

Input: (
Result: [ REJECTED ]
```

```
Input: (a, (b, c), [d])
Result: [ ACCEPTED ]

Input: (
Result: [ REJECTED ]

Input: )
Result: [ REJECTED ]

Input: print("Hello World!"
Result: [ REJECTED ]

Input: (a,
Result: [ REJECTED ]

Input: (a b)
Result: [ REJECTED ]

Input: "hello
Result: [ REJECTED ]

Input: [a)
Result: [ REJECTED ]

Input: a, b
Result: [ REJECTED ]

Input: (a,,b)
Result: [ REJECTED ]

Input:
Result: [ ACCEPTED ]
hyuse@Hyuse:~/Desktop/Automata/Automata Simulations$
```