Practice Question on LEX-YACC

Consider the following grammar. Expressions generated by this grammar are called *strict* if the parentheses used strictly alternate. Thus an expression is *strict* if one of the following is true

a. An integer is always *strict*.

b. E_1 + E_2 is strict if both E_1 and E_2 are *strict*

c. **(** E **)** is strict is (i) E is *strict* and (ii) if E has parenthesized sub-expressions then all outer most parentheses inside E are **[** … **]**.

d. **[** E **]** is strict is (i) E is *strict* and (ii) if E has parenthesized sub-expressions then all outer most parentheses inside E are **(** … **)**

The grammar: E → **integer** | **(** E **)** | **[** E **]** | E **+** E

Write a LEX and YACC specification files that generates a program that can determine whether an input expression is *strict* of not.