1. In some programming languages a quote inside a string is represented by two quotes, for example “a”“b” represents the string: letter a, quote, letter b, “hi” represents the string: letter h, letter i, and “” represents the empty string.

a) Create an NFA defining such strings (assume lower case letters and quotes only).

b) Transform your NFA into a DFA.
2. We expand the expression grammar with an assignment statement:

1: S → id = E  
2: E → T E'  
3: E' → + T E'  
4: E' → ε  
5: T → F T'  
6: T' → * F T'  
7: T' → ε  
8: F → id  
9: F → (E)

a) Augment the grammar and create nullable, first and follow

b) Produce the LL(1) Predictive Parse Table for the augmented grammar.

3. Given an augmented, grammar for bracket nests:
0: $S' \to S \,$  
1: $S \to ( \, S \, ) \, S$  
2: $S \to id$

a) Show a rightmost derivation of $( \, id \, ) \, id$$

b) Produce the LR(0) State Diagram for the above grammar

c) Produce the LR(0) parse table.