This exam has 2 questions, for a total of 10 points.

1. [5 points] Using the grammar that you used for assignment 2, draw the parse tree and the abstract syntax tree for the following Python program.
   
   ```python
   print - 1 + input()
   ```

   **Solution:** Here’s the parse tree in outline form.

   ```
   module
   | |-stmt_list
   | | |-stmt
   | | | | |-"print"
   | | | |-expr
   | | | | | |-expr
   | | | | | | | |-"-
   | | | | | | | |-expr
   | | | | | | | | |-"+
   | | | | | |-INT
   | | | | |-"="
   | | |-expr
   | | | |-INPUT
   | | | | |-"("
   | | | | |-")"
   | |-stmt_list
   | |-empty
   ```

   Here’s the AST

   ```
   Module(None, Stmt([Printnl([Add((UnarySub(Const(1)), CallFunc(Name('input'), [], None, None))], None))]))
   ```
2. [5 points] Write down or draw the LALR(1) parse table (state transition diagram) for the following grammar. Resolve shift-reduce errors in favor of shifting. In each state, list the items (grammar rules with dots) for that state and the actions (shift, goto, and reduce).

(1) start ::= match
(2) match ::= "a" "b"
(3) match ::= "a" match "b"

Solution:

state 0
  start ::= . match
  match ::= . "a" "b"
  match ::= . "a" match "b"
  on "a" shift to state 1
  on match goto state 2

state 1
  match ::= . "a" . "b"
  match ::= . "a" . match "b"
  match ::= . "a" "b"
  match ::= . "a" match "b"
  on "a" shift to state 1
  on "b" shift to state 3
  on match goto state 4

state 2, accept
  start ::= match .

state 3
  match ::= . "a" "b" .
  on end, "a", or "b" reduce by rule 2

state 4
  match ::= . "a" match . "b"
  on "b" shift to state 5

state 5
  match ::= . "a" match "b" .
  on end, "a", or "b" reduce by rule 3