CSI 30 Standard Pseudocode Terminology

(1) **Expression**
  (a) Form:

  Arithmetic expression using numbers, variables, and any of the following operations: Addition, Subtraction, Multiplication, Division, Exponentiation, Step Functions; *and possibly other operations specified in the problem*

  (b) Example: $3 + (2 \cdot n)^x$

(2) **Lists** (a datatype)
  (a) Form:
  - Referencing: [List name][[positive integer]]
  - Length: Len([List name])

  (b) Example:
  - Referencing: L[3] (gives the third element of the list)
  - Length: Len(L) has value 4 (if L is a length 4 list)

(3) **Assignment**
  (a) Form: $[variable] := [expression]$
  (b) Example: $x := 2 + n$

(4) **Procedure**
  (a) Form: procedure [name] ([list of variables] : [requirements on variables])
  (b) Example: procedure Bronx (x, y, z : integers such that $x, y < z$)

(5) **Comments**
  (a) Form: $\{[comments]\}$
  (b) Example: { these comments have no effect on the algorithm!! }

(6) **Propositions**
  (a) Form: [expression of propositional logic where basic propositions may use numbers, number variables, and the following relations on numbers $<, >, \geq, \leq, =$]
  (b) Example: $((x > 2) \wedge (y > 3)) \vee (x = -1)$

(7) **Conditional Constructions**
  (a) Form: If [proposition] then { [algorithm] }
  (b) Example:

  If $(x < 100)$ then
     { $x := x + 2$
      $y := 0$ }

  (a) Form: If [proposition] then { [algorithm] } else { [algorithm] }
  (b) Example:

  If $(x < 100)$ then
    $x := x + 2$
    $y := 0$
  else
    $y := 1$

(8) **For Loops**
  (a) Form: For [variable] := [start] to [end] by [step size] { [Algorithm] }
  (b) Example:

  For k := 2 to 8 by 2
    $x := x + k$

(9) **While Loops**
  (a) Form: While ([proposition]) { [Algorithm] }
  (b) Example:

  While ($k < 9$)
    $x := x + k$
    $k := k + 1$

(10) **Subprocedure** (only used ones specifically allowed by the problem!)
    (a) Form: [subprocedure name]([input to subprocedure])
    (b) Example: prime(28)

(11) **Return**
    (a) Form: Return ([expression])
    (b) Example: Return$(x/2)$