1)

a) No. Parsing C requires resolving ambiguities that are dependent on context (where they appear).

b) A language with an ambiguous grammar will have more than one parse tree. This does not preclude the language having some unambiguous grammar with only one parse tree so yes, A language can have both at the same time.

c) An ambiguous language has more than one parse tree and so cannot be LL or LR by definition.

d) LR(1) is more powerful than LL(1) because LR(1) parsing uses a bottom-up approach while LL(1) uses a top-down approach. The bottom-up approach allows for evaluation of successive input tokens (shifting in an effort to find a proper m non-terminal representation (reducing). This gives LR(1) parsers the ability to parse more grammars than LL(1).

1. (16 points) Answer the following questions about languages and grammars.

   a) Is the C programming language a context free language? Why or why not?

   b) Can a language have ambiguous and unambiguous grammars at the same time? Why or why not?

   c) Explain in your own words why ambiguous grammars can never be LL or LR.

   d) Explain in your own words why LR(1) is more powerful than LL(1).

2. (12 points) Answer the following questions about context free and regular grammars.

   a) Write a grammar for non-empty strings with matching quotes where $\sum$ = {a, b, '}. E.g. 'aba', aba''bb, aa'a'a'a'.

   b) Write a grammar for non-empty strings with matching parentheses $\sum$ = {a, b, (,)}. E.g. (aba), aba()bb, aa(a(a)a).

   c) Is the language expressed in a) a regular language? If not, explain why not. If so, modify grammar so that it is a regular grammar (if not already).

   d) Is the language expressed in b) a regular language? If not, explain why not. If so, modify grammar so that it is a regular grammar (if not already).

3. (10 points) Given the following grammar, construct the *First* sets for each RHS (right hand side) and *Follow* sets for each non-terminal symbol.
   $A \rightarrow BAc \mid FE$
   $B \rightarrow bEF \mid g$
   $E \rightarrow e \mid \varepsilon$
   $F \rightarrow f \mid EH$
   $H \rightarrow h$

4. (16 points) For each of the below grammars, answer the following questions:
   1) Is the grammar LL(1)? If so, write the LL(1) parse table. If not, point out the conflict using First and Follow sets.
   2) Is the grammar LL(k)? If so, show how extra lookahead resolves the conflict. If not, point out the unresolved conflict using First and Follow sets.
   ⋆ *Note: LL(k) is LL with a finite amount of lookahead.*
   3) Is the grammar ambiguous? If yes, find the input that produces two or more left-derivations.
   ⋆ *Note LL(1) ⊂ LL(k) ⊂ L(Unambiguous). So, (2) needs answering only if (1) is false. (3) needs answering only if both (1) and (2) are false.*

   a) $S \rightarrow [S \mid A$
      $A \rightarrow [A] \mid \varepsilon$

   b) $S \rightarrow ABc$
      $A \rightarrow a \mid \varepsilon$
      $B \rightarrow b \mid \varepsilon$

   c) $S \rightarrow ABBA$
      $A \rightarrow a \mid \varepsilon$
      $B \rightarrow b \mid \varepsilon$

   d) $S \rightarrow aAbc | bAc$
      $A \rightarrow b \mid \varepsilon$

5. (16 points) Given the following grammar, answer the below questions:
   $E \rightarrow E + E \mid id$

   a) Write a new grammar after performing left-recursion removal.

   b) Is the grammar in a) LL(1)? If not, point out the conflict using First and Follow sets.

   c) Modify the original grammar such that the + operator is left associative, *and then* perform left-recursion removal. Write the new grammar.

   d) Is the grammar in c) LL(1)? If not, point out the conflict using First and Follow sets

6. (30 points) For each of the below grammars, answer the following questions:
   (1) Is the grammar SLR(1)? If so, draw a DFA and the corresponding parse table. If not, point out the conflict.
   (2) Is the grammar LALR(1)? If so, show how the lookahead component resolves the conflict. If not, show why it is not resolved.
   (3) Is the grammar LR(1)? If so, show how state splitting resolves the conflict. If not, show why it is not resolved.
   ⋆ *Note SLR(1) ⊂ LALR(1) ⊂ LR(1). So, (2) needs answering only if (1) is false. (3) needs answering only if both (1) and (2) are false.*

   a) (10 points) Σ={v, =, ;, +, (,)}.
      $\mathbf{S} \rightarrow v\ =\ \mathbf{A};$
      $\mathbf{A} \rightarrow \mathbf{P}\ \mathbf{E}$
      $\mathbf{P} \rightarrow \mathbf{P}\ v\ =\ \mid \varepsilon$
      $\mathbf{E} \rightarrow \mathbf{E}\ +\ \mathbf{T} \mid \mathbf{T}$
      $\mathbf{T} \rightarrow v \mid (\ \mathbf{A}\ )$

   b) (10 points) Σ={a, b, c}.
      $\mathbf{S} \rightarrow b\mathbf{A}b \mid \mathbf{A}c \mid ab$
      $\mathbf{A} \rightarrow a$

   c) (10 points) Σ={a, b, c, d}.
      $\mathbf{S} \rightarrow \mathbf{A}a \mid b\mathbf{A}c \mid \mathbf{B}c \mid b\mathbf{B}a$
      $\mathbf{A} \rightarrow d$
      $\mathbf{B} \rightarrow d$

2)

a) $[\text{"}aa^*\text{"}]^*[\text{"}bb^*\text{"}]^*$ $\Rightarrow$

$S \to AC$
$A \to \text{"}aB\text{"}$
$B \to aB \mid \varepsilon$
$C \to \text{"}bD\text{"}$
$D \to bD \mid \varepsilon$

b) $[(aa^*)]^*[(bb^*)]^*$ $\Rightarrow$

$S \to AC$
$A \to (aB)$
$B \to aB \mid \varepsilon$
$C \to (bD)$
$D \to bD \mid \varepsilon$

c) No, all rules are not of the form 1) $A \to \varepsilon$ 2) $A \to a$ 3) $A \to B$ or 4) $A \to aB$. The language cannot be represented with a DFA since we must "keep track" of how many quotes we've seen.

d) No, all rules are not of the form 1) $A \to \varepsilon$ 2) $A \to a$ 3) $A \to B$ or 4) $A \to aB$. The language cannot be represented with a DFA since we must "keep track" of how many parentheses of a certain type we've seen.

3)

First sets:

$A = \{b, g, f, e, \varepsilon, h\}$

$E = \{e, \varepsilon\}$

$H = \{h\}$

$B = \{b, g\}$

$F = \{f, e, \varepsilon, h\}$

Follow sets:

A = {\$, c}   B = {b, g, f, e, h}   G = {h, f, e, \$, c}

F = {e, b, g, f, h, \$, c}   H = {e, b, g, f, h, \$, c}

4)

a) First sets: A = {[, ε} S = {[, ε}

Follow sets: S = {\$}, A = {], \$}

	[	]	\$
A			
S	[S/[A		

No, this is not LL(1) since we have multiple cell entries

b) First sets: S = {c, a, ε, b}, A = {a, ε}, B = {b, ε}

Follow sets: S = {\$}, A = {c, b}, B = {c}

	a	b	c	\$	
A	a	ε	ε		
S	ABc			ε	
B		b	ε		

Yes, this is LL(1)

c) First sets: S = {a, b, ε}, A = {a, ε}, B = {b, ε}

Follow sets: S = {\$}, A = {b, a, \$}, B = {b, a, \$}

	a	b	\$		
A	a/ε	ε			
S	ABBA				
B		b/ε			

No, this is not LL(1)

d) First sets: $S = \{a, b\}$, $A = \{b, \varepsilon\}$

Follow sets: $S = \{\$\}$, $A = \{b, c\}$

	a	b	c	$	
A		b / ε	ε		
S	aAbc	bAc			
B					

No, this is not LL(1)

5) $E \rightarrow E + E \mid id$

a) $E \rightarrow id\ E'$

$E' \rightarrow \varepsilon \mid + E E'$

b) First sets: $E = \{id\}$, $E' = \{\varepsilon, +\}$

Follow sets: $E = \{\$, +\}$, $E' = \{\$, +\}$

	id	+	$		
E	id E'				
E'		+EE'/ε			

No, this is not LL(1)

c) $E \rightarrow + E E \mid id$

$\Rightarrow E \rightarrow + E E'$

$E' \rightarrow E \mid \varepsilon$

$E \rightarrow id$

d) First sets: $E = \{t, id\}$, $E' = \{\Sigma, +, id\}$

Follow sets: $E = \{\$, +, id\}$, $E' = \{\$, +, id\}$

	id	+	\$		
E	'	+EE'			
E'	E/id				

No, this is not LL(1)

6)

a) (10 points) Σ={v, =, ;, +, (,)}.

$\mathbf{S} \rightarrow v\ =\ \mathbf{A};$
$\mathbf{A} \rightarrow \mathbf{P}\ \mathbf{E}$
$\mathbf{P} \rightarrow \mathbf{P}\ v\ =\ \mid \varepsilon$
$\mathbf{E} \rightarrow \mathbf{E}\ +\ \mathbf{T} \mid \mathbf{T}$
$\mathbf{T} \rightarrow v \mid (\ \mathbf{A}\ )$

b) (10 points) Σ={a, b, c}.

$\mathbf{S} \rightarrow b\mathbf{A}b \mid \mathbf{A}c \mid ab$
$\mathbf{A} \rightarrow a$

c) (10 points) Σ={a, b, c, d}.

$\mathbf{S} \rightarrow \mathbf{A}a \mid b\mathbf{A}c \mid \mathbf{B}c \mid b\mathbf{B}a$
$\mathbf{A} \rightarrow d$
$\mathbf{B} \rightarrow d$

a) 1) Yes

b) 1) No 2) Yes

Firsts: $S = \{b, a\}$, $A = \{a\}$

Follow: $S = \{\$\}$, $A = \{b, c\}$

c) 1) Yes

Firsts: $S = \{b, d\}$, $A = \{d\}$, $B = \{d\}$

Follow: $S = \{\$\}$, $A = \{a, c\}$, $B = \{c, a\}$