Question 1 of 9

A context-free grammar has (check all that apply)

- A. at least a non-terminal symbol in the left hand side of any production
- B. just only one non-terminal symbol in the left hand side of any production
- C. unlimited number of terminal and non-terminal symbols in the right hand side of any production
- D. at most one non-terminal symbol in the right hand side of any production

Question 2 of 9

Given the following grammar

\[ S \rightarrow aSbS | c \]

which of the following derivation processes are valid? Check all that apply

- A. \[ S \rightarrow aSbS \Rightarrow acbc \]
- B. \[ S \rightarrow aSbS \Rightarrow aaaSbSbS \Rightarrow aacbcSbS \Rightarrow aacbc \]
- C. \[ S \rightarrow aSbS \Rightarrow acbS \Rightarrow aacbc \]
- D. \[ S \rightarrow aSbS \Rightarrow acbS \Rightarrow acbc \]

Question 3 of 9

Given the following grammar

\[ S \rightarrow AB \]
\[ A \rightarrow aAa | \epsilon \]
\[ B \rightarrow bB | b \]

which of the following derivation processes are valid? Check all that apply

- A. \[ S \Rightarrow AB \Rightarrow B \Rightarrow b \]
- B. \[ S \Rightarrow AB \Rightarrow AbB \Rightarrow Abb \Rightarrow aAabb \Rightarrow aaaabbbb \]
- C. \[ S \Rightarrow AB \Rightarrow AbB \Rightarrow aAabB \Rightarrow aabb \]
- D. \[ S \Rightarrow AB \Rightarrow aAaB \Rightarrow aaAaaB \Rightarrow aaAaab \Rightarrow aaabab \]

Question 4 of 9

Which of the following grammars are ambiguous? Check all that apply
A. $S \to SS \mid a \mid b$
B. $A \to AcA \mid b$
C. $S \to aS \mid bS \mid a \mid b$
D. $A \to AcB \mid B$

$B \to yBz \mid t \mid xB$

Question 5 of 9
1.0 Points
Based on the following grammar, which of the following parse trees are valid for input aabb? Check all that apply:

$S \to AB$
$A \to aAa \mid \epsilon$
$B \to bB \mid b$

- A.
- B.
Question 6 of 9

Assume that an expression in a language is defined as follows

<exp> → <term> ASSIGN <exp>
| <term>
<term> → <term> EXPONENT <fact>
| <term> ADDOP <fact>
| <fact>
<fact> → <opeand> RELOP <operand>
| <operand>
<operand> → LB <exp> RB
| ID

Which of the following sentences are **correct**? Check all that apply
A. ASSIGN is left-associated
B. ASSIGN is non-associated
C. ASSIGN is right-associated
D. EXPONENT is left-associated
E. EXPONENT is non-associated
F. EXPONENT is right-associated
G. ADDOP is left-associated
H. ADDOP is non-associated
I. ADDOP is right-associated
J. RELOP is left-associated
K. RELOP is non-associated
L. RELOP is right-associated

Question 7 of 9 1.5 Points
Assume that an expression in a language is defined as follows

\[
<\text{exp}> \rightarrow <\text{term}> \text{ASSIGN} <\text{exp}>
\]

\[
| <\text{term}>
\]

\[
<\text{term}> \rightarrow <\text{term}> \text{EXPONENT} <\text{fact}>
\]

\[
| <\text{term}> \text{ADDOP} <\text{fact}>
\]

\[
| <\text{fact}>
\]

\[
<\text{fact}> \rightarrow <\text{opeand}> \text{RELOP} <\text{operand}>
\]

\[
| <\text{opeand}>
\]

\[
<\text{opeand}> \rightarrow \text{LB} <\text{exp}> \text{RB}
\]

\[
| \text{ID}
\]

Which of the following sentences are correct? Check all that apply

A. ASSIGN has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
B. ASSIGN has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
C. ASSIGN has equal precedence to another operator
D. EXPONENT has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP

E. EXPONENT has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP

F. EXPONENT has equal precedence to another operator

G. ADDOP has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP

H. ADDOP has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP

I. ADDOP has equal precedence to another operator

J. RELOP has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP

K. RELOP has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP

L. RELOP has equal precedence to another operator

Question 8 of 9

Assume that an expression in a language is defined as follows:

<exp> → <term> ASSIGN <exp> | <term>
<term> → <term> EXPONENT <fact> | <term> ADDOP <fact> | <fact>
<fact> → <operand> RELOP <operand> | <operand>
<operand> → LB <exp> RB | ID

Let ID be token of identifiers, ASSIGN of '='; EXPONENT of '^'; ADDOP of '*'; RELOP of '>'; LB of '{' and RB of '}'

Which of the following trees is the abstract syntax tree of expression \( a = b + c > d \cdot e \)?

A. \( \frac{a}{=} \frac{b}{a} + \frac{c}{d} \cdot \frac{e}{e} \)
Assume that an expression in a language is defined as follows:

\(<\text{exp}\> \rightarrow <\text{term}\> \text{ ASSIGN } <\text{exp}\> | <\text{term}\>

\(<\text{term}\> \rightarrow <\text{term}\> \text{ EXPONENT } <\text{fact}\> | <\text{term}\> \text{ ADDOP } <\text{fact}\> | <\text{fact}\>

\(<\text{fact}\> \rightarrow <\text{opeand}\> \text{ RELOP } <\text{operand}\> | <\text{operand}\>
Let ID be token of identifiers, ASSIGN of '=' , EXPONENT of '^' , ADDOP of '+' , RELOP of '>' , LB of '(' and RB of ')'.

Association and precedence help to reduce the brackets in an expression. For example, if operator - is left-associated, \((a - b) - c\) can be rewritten as \(a - b - c\) with the same meaning. With the above grammar, which of the following expressions are equivalence to expression \((a = ((b + c) > ((c = (a + b)) ^ d)))\) ? Check all that apply

- A. \(a = b + c > c = a + b ^ d\)
- B. \(a = b + c > (c = a + b) ^ d\)
- C. \(a = b + c > ((c = a + b) ^ d)\)
- D. \(a = (b + c) > (c = a + b ^ d)\)
- E. \((a = b + c) > (c = a + b ^ d)\)
- F. \(a = (b + c) > ((c = a + b) ^ d)\)
- G. \(a = ((b + c) > ((c = a + b) ^ d))\)
- H. \((a + c) > (c = (a + b) ^ d))\)