1. Which of the following statements is false?
(A) An unambiguous grammar has same left most and right most derivation
(B) An LL(1) parser is a top-down parser
(C) LALR is more powerful than SLR
(D) An ambiguous grammar can never be LR (K) for any k

SOLUTION
So (A) & (C) are true.
An ambiguous grammar can’t be LR (K)
So option (A) is false since an unambiguous grammar has unique right most derivation
& left most derivations but both are not same. Hence (A) is correct option

2. Dynamic linking can cause security concerns because
(A) Security is dynamic
(B) The path for searching dynamic libraries is not known till run time.
(C) Linking is insecure
(D) Cryptographic procedures are not available for dynamic linking

SOLUTION
Dynamic linking is type of linking in which libraries required by the program are linked during run
time. But at this time cryptographic procedures are not available, so make this process insecure.
Hence (D) is correct option.

3. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?
(A) Removing left recursion alone
(B) Factoring the grammar alone
(C) Removing left recursion and factoring the grammar
(D) None of this

SOLUTION
If a grammar has left recursion & left factoring then it is ambiguous. So to convert a CFG to LL(1)
grammar both removal of left recursion & left factoring need to be done.
Hence (C) is correct option.

4. Assume that the SLR parser for a grammar G has $n_1$ states and the LALR parser for G has $n_2$
states. The relationship between $n_1$ and $n_2$ is
(A) \(n_1\) is necessarily less than \(n_2\)
(B) \(n_1\) is necessarily equal to \(n_2\)
(C) \(n_1\) is necessarily greater than \(n_2\)
(D) None of the above

**SOLUTION**

SLR pursue is less range of context free languages than LALR but still both \(n_1\) & \(n_2\) are same for SLR & LALR respectively.
Hence (B) is correct option.

5. In a bottom-up evaluation of a syntax directed definition, inherited attributes can
(A) always be evaluated
(B) be evaluated if the definition is L-attributed
(C) be evaluated only if the definition has synthesized attributes
(D) never be evaluated

**SOLUTION**

Every \(S\) (synthesized) -attributed definitions is \(L\)-attributed. So in a bottom-up evaluation of \(SDD\) inherited attributes can be evaluated only if the definition has synthesized attributes.
Hence (C) is correct option.

6. Which of the following statements is FALSE?
(A) In statically typed language, each variable in a program has a fixed type
(B) In up-typed languages, values do not have any types
(C) In dynamically typed languages, variables have no types
(D) In all statically typed languages, each variable in a program is associated with values of only a single type during the execution of the program

**SOLUTION**

(1) True for statically typed languages where each variable has fixed type. Similarly (4) is also correct.
(2) True, in un-typed languages types of values are not defined.
But option (C) is false, since in dynamically typed language variables have dynamically changing types but not that they have no type.
Hence (C) is correct option.

7. Consider the grammar shown below
\[ S \rightarrow E \bigcup SS \bigcup \alpha \]
8. Consider the grammar shown below.

\[
S \rightarrow CC \\
C \rightarrow eC \mid d
\]

The grammar is

(A) LL (1)  
(B) SLR (1) but not LL (1)  
(C) LALR (1) but not SLR (1)  
(D) LR (1) but not LALR (1)

**SOLUTION:**

Given grammar

\[
S \rightarrow CC \\
C \rightarrow eC \mid d
\]

it can’t be LL since \( C \rightarrow eC \) is recursive. \( LR(1) \) also known as CLR parser, and every CF grammar is CLR grammar.

So (A) is false but (C) & (D) can be correct.

This grammar is CLR and also reducible to LALR without any conflicts. So (D) is false.

Only need to check for SLR(1) or LR(0)

This grammar is not SLR.

Hence (C) is correct option

9. Consider the translation scheme shown below

\[
S \rightarrow TR
\]
Here num is a token that represents an integer and num.val represents the corresponding integer value. For an input string ‘9 + 5 + 2’, this translation scheme will print

(A) 9 + 5 + 2  
(B) 9 5 + 2 +  
(C) 9 5 2 ++  
(D) ++ 9 5 2

SOLUTION

Given string 9 + 5 + 2

So ++ 952 is printed.

Hence (D) is correct option.

10. Consider the syntax directed definition shown below

S " id: = E
      "
      newtemp ();
      gen(t . place . place;);
      "
      .place t}
      .place .place;)
      "

Here, gen is a function that generates the output code, and newtemp is a function that returns the name of a new temporary variable on every call. Assume that t1’s are the temporary variable names generated by newtemp.

For the statement ‘X: = Y + Z’, the 3-address code sequence generated by this definition is

(A) X = Y + Z
(B) t1 = Y + Z; X t1
(C) t1 = Y; t2 = t1 + Z; X = t2
(D) t1 = Y; t2 = Z; t3 + t2; X = t3
SOLUTION

In 3-address code we use temporary variables to reduce complex instructions so here
\[ t_1 = Y \]
\[ t_2 = Z \]
\[ t_3 = t_1 + t_2 \]
\[ x = t_3 \]
Hence (D) is correct option.

Solve the problems and choose the correct answers.

The following program fragment is written in a programming language that allows variables and does not allow nested declarations of functions.

```plaintext
global int
void
    int i
    print i
    print
}
main () { (i   ) }
```

11. If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are
(A) 115, 220
(B) 25, 220
(C) 25, 15
(D) 115, 105

SOLUTION

In static scoping the variables are initialized at compile time only
So \( i = 100 \) & \( j = 5 \)
\[ P(i + j) = P(100 + 5) = P(105) \]
So
\[ x = 105 \]
\[ x + 10 = 105 + 10 = 115 \]
So 115 & 105 will be printed.
Hence (D) is correct option.

12. If the programming language uses dynamic scoping and call by name parameter passing mechanism, the values printed by the above program are
(A) 115, 220  (B) 25, 220  
(C) 25, 15  (D) 115, 105

**SOLUTION**

In dynamic scoping, the local values are considered & variables are initialized at run time. Since
\[ x = i + j \] & in \( P(x) \)
\[ i = 200 \] & \( j = 20 \) \( x = 200 + 20 = 220 \)
& printing \((x + 10)\)
\[ q = i + j + 10 \]
\[ = 10 + 5 + 10 = 25 \]

Hence (B) is correct option

13. Consider the following class definitions in a hypothetical object oriented language that supports inheritance and uses dynamic binding. The language should not be assumed to be either Java or C++, though the syntax is similar.

```java
class P {  
    void f(int i) {  
        print (i);  
    }  
}  

class Q subclass of P {  
    void f (int i) {  
        print ( i);  
    }  
}  
```

Now consider the following program fragment:

\( P \ x = \text{new } Q(); \)
\( Q \ y = \text{new } Q(); \)
\( P \ z = \text{new } Q(); \)
\( x.f(1);((P) \ y).f(1);z.f(1); \)

Here \((P) \ y\) denotes a typecast of \( y \) to \( P \). The output produced by executing the above program fragment will be

(A) 1 2 1  (B) 2 1 1  
(C) 2 1 2  (D) 2 2 2
1. \( P_x = newQ() \);
2. \( Q_y = newQ() \);
3. \( P_z = newQ() \);
4. \( x : f(1); \) print 2 # \( i = 2 \)
5. \( (P \ y) : f(1) \);
6. \( z : f(1) \) print 2 # \( i = 2 \)

but line 5. will print 2 because typecast to parent class can’t prevent over ridding. So function \( f(1) \) of class \( Q \) will be called not \( f(1) \) of class \( P \).

Hence (D) is correct option.

14. Which of the following is NOT an advantage of using shared, dynamically linked libraries as opposed to using statically linked libraries?

(A) Smaller sizes of executable
(B) Lesser overall page fault rate in the system
(C) Faster program startup
(D) Existing programs need not be re-linked to take advantage of newer versions of libraries

SOLUTION

The advantages of shared dynamically linked libraries include.
(A) smaller size of executable since less data
(B) lesser overall page fault rate.
(C) No need for re-linking if newer versions of libraries are there.

But since compilation time doesn’t include linking so a long linking time required during runtime in DLL’s so slow startup.

Hence (C) is correct option.

15. Which of the following grammar rules violate the requirements of an operator grammar? \( P, Q, R \) are non-terminals, and \( r, s, t \) are terminals

(i) \( P \ " QR \)  
(ii) \( P \ " Q s R \)
(iii) \( P \ " \varepsilon \)  
(iv) \( P \ " Q t R r \)

(A) (i) only  
(B) (i) and (iii) only
(C) (ii) and (iii) only  
(D) (iii) and (iv) only

SOLUTION
(I) \( P " QR \) is not possible since two \( NT \) should include one operator as Terminal.
(II) Correct
(III) Again incorrect. (IV) Correct.
Hence (B) is correct option.

16. Consider a program \( P \) that consists of two source modules \( M_1 \) and \( M_2 \)
contained in two different files. If \( M_1 \) contains a reference to a function
defined in \( M_2 \), the reference will be resolved at
(A) Edit-time  (B) Compile-time
(C) Link-time  (D) Load-time

SOLUTION
The two modules needed to be linked since definition exist \& \( M_2 \) \& \( M_1 \) refers it. So during linking
phase \( M_1 \) links to \( M_2 \).
Hence (C) is correct option.

17. Consider the grammar rule \( E " E_1 - E_2 \) for arithmetic expressions. The code generated is
targeted to a CPU having a single user register. The subtraction operation requires the first operand
to be in the register. If \( E_1 \) and \( E_2 \) do not have any common sub expression, in order to get the
shortest possible code
(A) \( E_1 \) should be evaluated first
(B) \( E_2 \) should be evaluated first
(C) Evaluation of \( E_1 \) and \( E_2 \) should necessarily be interleaved
(D) Order of evaluation of \( E_1 \) and \( E_2 \) is of no consequence

SOLUTION
\( E_1 \) is to be kept in accumulator \& accumulator is required for operations to evaluate \( E_2 \) also. So \( E_2 \)
should be evaluated first \& then \( E_1 \), so finally \( E_1 \) will be in accumulator, otherwise need to use
move \& load instructions.
Hence (B) is correct option.

18. Consider the grammar with the following translation rules and \( E \) as the start symbol.
\[
E \ " E_1 \ #T \quad .value = .value * .value \\
\mid \quad .value = .value \\
\ " \quad .value = .value + .value \\
\ "num \quad .value = num.value \\
\]
Compute $E$ value for the root of the parse tree for the expression: $2 \# 3 \# 5 \# 6 \& 4$.

(A) 200  
(B) 180  
(C) 160  
(D) 40

**SOLUTION**

The parse tree would be.

```
E
  /\  
E #  
  /\  
T #  
  /\  
T &  
  /\  
F    
```

Now we evaluate bottom up

```
T
  /\  
T &  
  /\  
F    
num (3)  
num (5)
```

$3 + 5 = 8$

```
T
  /\  
T &  
  /\  
F    
num (6)  
num (4)
```

$6 + 4 = 10$
LL " left to right left most derivation no ambiguity should be there

SLR or LR(0) L to R reverse right sentential form create LR(0) items.

CLR or LR(1) create LR(1) items no bound

LALR reduced CLR if while reducing any conflict found then not LALR

Hence (C) is correct option.

19. The grammar $A \rightarrow AA | (A) | \epsilon$ is not suitable for predictive-parsing
because the grammar is
(A) ambiguous    (B) Left-recursive
(C) right-recursion (D) an operator-grammar

SOLUTION

The grammar is definitely left & right recursive but it is not suitable for predictive parsing because it is ambiguous.
Hence (A) is correct option.

20. Consider the grammar $E \rightarrow E + n \mid E \# n \mid n$

For a sentence $n + n$, the handles in the right-sentential form of the reduction are
(A) $n$, $E + n$ and $E + n \# n$
(B) $n$, $E + n$ and $E + E \# n$
(C) $n$, $n + n$ and $n + n \# n$
(D) $n$, $E + n$ and $E \# n$

SOLUTION

Given grammar

\[
E \rightarrow E + n \\
E \rightarrow E \# n \\
E \rightarrow n
\]

String = $n + n \# n$

Right sentential so right most non terminal will be used.

\[
E \rightarrow E \# n \qquad \{E \rightarrow E \# n\} \\
E + n \# n \qquad \{E \rightarrow E + n\} \\
n + n \# n \qquad \{E \rightarrow n\}
\]

So during reduction the order is reverse.
So
\[
\{E \rightarrow n, E \rightarrow E + n, E \rightarrow E \# n\}
\]

Hence (D) is correct option.

21. Consider the grammar $S \rightarrow (S) \mid a$

Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar $n_1$ $n_2$ and $n_3$ respectively. The following relationship holds good
(A) $n_1 < n_2 < n_3$    (B) $n_1 = n_3 < n_2$
(C) $n_1 = n_2 = n_3$    (D) $n_1 \neq n_3 \neq n_2$

SOLUTION

The no. of states for SLR(1) & LALR(1) are equal so $n_1 = n_3$, but CLR(1) or LR(1) will have no. of
states greater than $LALR & LR(0)$ both.
Hence (B) is correct option.

22. Consider line number 3 of the following C-program.

```c
int main (){
    int 1, N;
    for (i = 1; i < N; i)  // * Line 3 *
}
```

Identify the compiler’s response about this line while creating the object-module
(A) No compilation error
(B) Only a lexical error
(C) Only syntactic errors
(D) Both lexical and syntactic errors

**SOLUTION**

There are no lexical errors for C because all the wrong spelled keywords would be considered as identifiers until the syntax is checked.
So the compiler would give syntax errors.
Hence (C) is correct option.

**Data for Q. 23 & 24 are given below.**

Solve the problems and choose the correct answers.

Consider the following expression grammar. The semantic rules for expression calculation are stared next to each grammar production.

```
E  "  number              Eval    number  val  
    E                      E .val   E .VAL    E .val
    E  #  E                 E .val   E .VAL    E .val

; 
```

23. The above grammar and the semantic rules are fed to a yacc tool (which is an LALR(1) parser generator) for parsing and evaluating arithmetic expressions. Which one of the following is true about the action of yacc for the given grammar?
(A) It detects recursion and eliminates recursion
(B) It detects reduce-reduce conflict, and resolves
(C) It detects shift-reduce conflict, and resolves the conflict in favor of a shift over a reduce action
(D) It detects shift-reduce conflict, and resolves the conflict in favor of a reduce over a shift action

SOLUTION

Yace tool is used to create a LALR(1) parser. This parser can detect the conflicts but to resolve the conflicts it actually prefers shift over reduce action. Hence (C) is correct option.

24. Assume the conflicts part (a) of this question are resolved and an LALR(1) parser is generated for parsing arithmetic expressions as per the given grammar. Consider an expression 3 # 2 + 1. What precedence and associativity properties does the generated parser realize?
(A) Equal precedence and left associativity; expression is evaluated to 7
(B) Equal precedence and right associativity, expression is evaluated to 9
(C) Precedence of ‘x’ is higher than that of ‘+’, and both operators are left associative; expression is evaluated to 7
(D) Precedence of ‘#’ is higher than that of ‘#’, and both operators are left associative; expression is evaluated to 9

SOLUTION

The grammar has equal precedence and it is also ambiguous. Since LALR(1) parser prefer shift over reduce so + operation will be executed here before ). 2 + 1 = 3 & 3 # 3 = 9 also the operators are right associative.
Hence (B) is correct option.

25. Consider the following grammar.

\[
\begin{align*}
S & \rightarrow S \ast E \\
S & \rightarrow E \\
E & \rightarrow F + E E \\
F & \rightarrow id
\end{align*}
\]

Consider the following LR(0) items corresponding to the grammar above.

(i) \( S \rightarrow S \ast .E \)
(ii) \( E \rightarrow F. + E \)
(iii) \( E \rightarrow F + .E \)

Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?
26. Consider the following grammar

\[
S \rightarrow FR \\
R \rightarrow S | F \rightarrow \epsilon | id
\]

In the predictive parser table, \( M \), of the grammar the entries \( M[S, id] \) and \( M[R,\epsilon] \) respectively

(A) \{ S " FR\} and \{ R " \epsilon \} \\
(B) \{ S " FR\} and \{ \} \\
(C) \{ S " FR\} and \{ R " \epsilon \} \\
(D) \{ F " id\} and \{ R " \epsilon \}

**SOLUTION**

The predictive parser table is given as.

<table>
<thead>
<tr>
<th>Non Terminal</th>
<th>)</th>
<th>id</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td>S &quot; FR</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>F &quot; id</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>R &quot;) S R &quot;!</td>
<td>R &quot;</td>
<td></td>
</tr>
</tbody>
</table>

So at

\[
M[S, id] = \{ S " FR\}
\]

Hence (A) is correct option.

27. Consider the following translation scheme.

\[
S \rightarrow ER \\
R \rightarrow E \{print(\text{'}\text{*}\text{'});R \mid f \\
E \rightarrow F + E \{print(\text{'}\text{+}\text{'}); \mid F \\
F \rightarrow (S) \mid \text{id}\{\text{print(id.value)}\}
\]

Here \( id \) is a taken that represents an integer and \( id.\) value represents the corresponding integer value.

For an input ‘2 * 3 + 4’, this translation scheme prints

(A) 2 * 3 + 4 \\
(B) 2 * 3 4 \\
(C) 2 3 * 4 + \\
(D) 2 3 4 + *
SOLUTION
Input string 2 ) 3 + 4

S " ER
FR
idR {print(2)}
 id)ER {print(3)}
id) F+ER {print(+) id}
id + ER {print(3) id id) id +id
So 2 )+ 3 4 are printed
Hence (B) is correct option.

28. Consider the following C code segment.
   for
   for
     if  i
       # i
     }
   }
}
Which one to the following false?
(A) The code contains loop-in variant computation
(B) There is scope of common sub-expression elimination in this code
(C) There is scope strength reduction in this code
(D) There is scope of dead code elimination in this code

SOLUTION
All the statements are true except option (D) since there is no dead code to get eliminated.
Hence (D) is correct option.

29. Which one of the following grammars generates the language

L = (a i b i | i ! j) ?

(A) S " AC | CB
   C " aCb | a | b
   A " aA | ε
   B " Bb | ε

(B) S " aS | Sb | a | b

(C) S " ACCB

(D) S " AC | CB

IT DEPARTMENT
GATE MATERIAL
The grammar
\[
S \rightarrow AC C B \\
C \rightarrow AC A \rightarrow aA a A \rightarrow aA a B \rightarrow bB b
\]
Consider string \textit{aaabb}
\[
S \rightarrow AC AaCb \\
AaaCbb \\
Aaabb aaabb
\]
But string \textit{aabb}
\[
S \rightarrow AC
\]
And this string is not derivable. Hence (D) is correct option.

30. In the correct grammar above, what is the length of the derivation (number of steps starting from \textit{S} to generate the string \textit{a}^l \textit{b}^m \textit{a} \textit{b} \textit{a}) with \textit{l} \neq \textit{m}?

(A) max (\textit{l}, \textit{m}) + 2
(B) \textit{l+m+2}
(C) \textit{l + m + 3}
(D) max (\textit{l}, \textit{m}) + 3

SOLUTION
It is very clear from the previous solution that the no. of steps required depend upon the no. of \textit{a}'s & \textit{b}'s which ever is higher & exceeds by 2 due to \textit{S \rightarrow AC CB & C}!
So max(\textit{l}, \textit{m}) + 2
Hence (A) is correct option.

31. Which one of the following is a top-down parser?
(A) Recursive descent parser
(B) Operator precedence parser
(C) An LR(k) parser
(D) An LALR(k) parser

SOLUTION
Clearly LR & LALR are not top down they are bottom up passers. Also not operator precedence parser.
But yes recursive descent parser is top down parser. Starts from start symbol & derives the terminal string.
Hence (A) is correct option.

32. Consider the grammar with non-terminals \( N = \{ S, C, S \} \), terminals \( T = \{ a, b, i, t, e \} \), with \( S \) as the start symbol, and the following of rules

\[
S \rightarrow i Ct SS_1 | a S_1 \quad e S \mid \varepsilon
\]

\( C \rightarrow b \)

The grammar is NOT LL(1) because:
(A) It is left recursive
(B) It is right recursive
(C) It is ambiguous
(D) It is not context-free

**SOLUTION**

The grammar has production

\( S \rightarrow i Ct SS_1 \) here the right hand side of grammar has the same symbol as left side. So the grammar is left recursive.

The grammar is not ambiguous. Hence (A) is correct option.

33. Consider the following two statements:

\( P \): Every regular grammar is LL(1)

\( Q \): Every regular set has LR(1) grammar

Which of the following is TRUE?

(A) Both \( P \) and \( Q \) are true
(B) \( P \) is true and \( Q \) is false
(C) \( P \) is false and \( Q \) is true
(D) Both \( P \) and \( Q \) are false

**SOLUTION**

LL(1) parsers can recognize the regular grammars also LL(1) is subset of LR(1) or CLR grammar so it also recognizes regular sets. So both accept regular grammar.

34. In a simplified computer the instructions are:

\( OP R_j, R_i \) – Performs \( R_j \) OP \( R_i \) and stores the result in register \( R_i \)

\( OP m, R_i \) – Performs val OP \( R_i \) and stores the result in \( R_i \), value denotes the content of memory location \( m \).

\( MCV m, R_i \) – Moves the content off memory location \( m \) to register \( R_i \).

\( MC V m, R_i, m \) – Moves the content of register \( R_i \) to memory location \( m \).

The computer has only two registers, and \( OP \) is either ADD or SUB. Consider the following basic
block:

\[ t_1 = a + b \]
\[ t_2 = c + d \]
\[ t_3 = e - t_2 \]
\[ t_4 = t_1 - t_2 \]

Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?

(A) 2  
(B) 3  
(C) 5  
(D) 6

**SOLUTION**

The operation sequence would be

\[ MOV a, R_1 \]
\[ ADD b, R_1 \] \{ \( R_1 = t_1 \) \}
\[ MOV c, R_2 \]
\[ ADD d, R_2 \] \{ \( R_2 = t_2 \) \}
\[ SUB e, R_2 \] \{ \( t_3 = e - R_2 = R_2 \) \}
\[ SUB R_1, R_2 \] \{ \( R_2 = t_4 \) \}
\[ MOV R_2, t_4 \] \{ finally in memory \}

Totally no. of move operation are 3

Hence (B) is correct option

Data for Q. 35 & 36 are given below.

Solve the problems and choose the correct answers.

Consider the CFG with \( \{ S, A, B \} \) as the non-terminal alphabet, \( \{ a, b \} \) as the terminal alphabet, \( S \) as the start symbol and the following set of production rules

\[
S \rightarrow bA \\
S \rightarrow aB \\
A \rightarrow a \\
A \rightarrow bS \\
A \rightarrow bAA \\
B \rightarrow b \\
B \rightarrow aS \\
B \rightarrow aBB
\]

35. Which of the following strings is generated by the grammar?

(A) \( aaaabb \)  
(B) \( aabbbb \)  
(C) \( aabbab \)  
(D) \( abbbba \)

**SOLUTION**
aabbab
S " aB
abajBB
abajBB
aabbSB
aabbAB
aabbab

Hence (C) is correct option.

36. For the correct answer string to Q. 9 how many derivation trees are there?
(A) 1
(B) 2
(C) 3
(D) 4

SOLUTION

For the derivation two trees are possible

So due to ambiguity 2 trees are possible

Hence (B) is correct option.

37. Which of the following describes a handle (as applicable to LR-parsing) appropriately?
(A) It is the position in a sentential form where the next shift or reduce operation will occur
(B) It is a non-terminal whose production will be used for reduction in the next step
(C) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.
(D) It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found

SOLUTION
Handles are the part of sentential form, & they are identified as the right side of any given production which will be used for reduction in the net step. Hence (D) is correct option.

38. Some code optimizations are carried out on the intermediate code because
(A) They enhance the portability of the compiler to other target processors
(B) Program analysis is name accurate on intermediate code than on machine code
(C) The information from data flow analysis cannot otherwise be used for optimization
(D) The information from the front end cannot otherwise be used for optimization

SOLUTION

Code optimizations are carried out on the intermediate code because program analysis is more accurate on intermediate code than on machine code.
Hence (B) is correct option.

39. Which of the following are true?
(i) A programming language option does not permit global variables of any kind and has no nesting of procedures/functions, but permits recursion can be implemented with static storage allocation
(ii) Multi-level access link (or display) arrangement is needed to arrange activation records-only if the programming language being implemented has nesting of procedures/function
(iii) Recursion in programming languages cannot be implemented with dynamic storage allocation
(iv) Nesting of procedures/functions and recursion require a dynamic heap allocation scheme and cannot be implemented with a stack-based allocation scheme for activation records
(v) Languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records

(A) (ii) and (v) only
(B) (i), (iii) and (iv) only
(C) (i), (ii) and (v)
(D) (ii), (iii) and (v) only

SOLUTION

I. Statement is false since global variables are required for recursions with static storage. This is due to unavailability of stack in static storage.
II. This is true
III. In dynamic allocation heap structure is used, so it is false. IV. False since recursion can be implemented.
V. Statement is completely true. So only II & V are true.
Hence (A) is correct option.
40. An LALR(1) parser for a grammar can have shift-reduce (S-R) conflicts if and only if
(A) The SLR(1) parser for G has S-R conflicts
(B) The LR(1) parser for G has S-R conflicts
(C) The LR(0) parser for G has S-R conflicts
(D) The LALR(1) parser for G has reduce-reduce conflicts

**SOLUTION**

LALR parser is reduced form of CLR or LR(1) parser, LALR parser uses the LR(1) items of CLR parser & of any shift reduce conflicts are there then it is due to LR(1) parser.

Hence (B) is correct option.

41. Which of the following statements are TRUE?

I. There exist parsing algorithms for some programming languages whose complex are less than $\Theta(n^3)$
II. A programming language which allows recursion can be implemented with static storage allocation
III. No L-attributed definition can be evaluated in the framework of bottom-up parsing
IV. Code improving transformations can be performed at both source language and intermediate code level

(A) I and II
(B) I and IV
(C) III and IV
(D) I, III and IV

**SOLUTION**

I. Statement is true since there are some parsers which take $O(n \log_2 n)$ time for parsing.

II. Completely false, since there is no use of stack which is required for recursion.

III. False

IV. True since both types of optimizations are applied

Hence (B) is correct option.

42. What data structure in a compiler is used for managing information about variables and their attributes?

(A) Abstract syntax tree
(B) Symbol table
(C) Semantic stack
(D) Parse table

**SOLUTION**
Symbol table is used for storing the information about variables and their attributes by compiler. Hence (B) is correct option.

43. Which languages necessarily need heap allocation in the runtime environment ?
(A) Those that support recursion
(B) Those that use dynamic scoping
(C) Those that allow dynamic data structure
(D) Those that use global variables

**SOLUTION**

Dynamic memory allocation is maintained by heap data structure. So to allow dynamic data structure heap is required. Hence (C) is correct option.

**EXERCISE QUESTIONS**

1. Given the language L- \{ ab, aa, baa \}, which of the following strings are in LG?
   1) abaabaaabaa
   2) aaaaabaaa
   3) baaaaabaaaab
   4) baaaaabaa
   (A) 1,2 and 3  (B) 2,3 and 4  (C) 1,2 and 4  (D) 1,3 and 4

   Answer: C

2. A simple two-pass assembler does which of the following in the first pass ?
   A. It allocates space for the literals
   B. It computes the total length of the program
   C. It builds the symbol table for the symbols and their values.
   D. All of these
   Answer: D

3. _________or scanning is the process where the stream of characters making up the source program is read from left to right and grouped into tokens.
   A. Lexical analysis
B. Diversion
C. Modeling
D. None of the above
Answer A

4. Which of the following is used for grouping of characters into tokens?
A parser
B Code optimization
C Code generator
D Lexical analyzer
Answer D

5. The lexical analyzer takes _______ as input and produces a stream of _______ as output.
A Source program, tokens
B Token, source program
C Either A and B
D None of the above
Answer A

6. The action of parsing the source program into proper syntactic classes is called
A Syntax analysis
B Lexical analysis
C Interpretation analysis
D General syntax analysis
Answer B

7. Task of the lexical analysis
A To parse the source program into the basic elements or tokens of the language
B To build a literal table and an identifier table
C To build a uniform symbol table
D All of these
Answer D

8. The output of lexical analyzer is
A A set of regular expressions
B Syntax tree
C Set of tokens
D Strings of character
Answer C

9. The output of a lexical analyzer is
A Machine code
10. Consider the program given below, in a block-structured pseudo-language with lexical scoping and nesting of procedures permitted.
Program main;
Var ...
Procedure A1;
Var ...
Call A2;
End A1
Procedure A2;
Var ...
Procedure A21;
Var ...
Call A1;
End A21
Call A21;
End A2
Call A1;
End main.
Consider the calling chain: Main -> A1 -> A2 -> A21 -> A1
The correct set of activation records along with their access links is given by

A   A
B   B
C   C
D   D
Answer D

11. Shift reduce parsers are
A Top down parser
B Bottom up parser
C May be top down or bottom up parser
D None of the above
Answer B

12. A bottom up parser generates
A Right most derivation
B Right most derivation in reverse
13. Inherited attribute is a natural choice in

A. Keeping track of variable declaration
B. Checking for the correct use of L values and R values
C. Both A and B
D. None of these

14. A top down parser generates

A. Right most derivation
B. Right most derivation in reverse
C. Left most derivation
D. Left most derivation in reverse

15. Recursive descent parsing is an example

A. Top down parsing
B. Bottom up parsing
C. Predictive parsing
D. None of the above

16. In operator precedence parsing, precedence relations are defined

A. For all pair of non terminals
B. For all pair of terminals
C. To delimit the handle
D. Only for a certain pair of terminals

17. A parser with the valid prefix property is advantageous because it

A. Detects error as soon as possible
B. Detects errors as and when they occur
C. Limits the amount of erroneous output passed to the text phase
D. All of these

18. The linker

A. is similar to interpreter
B. uses source code as its input
19. Which of the following is used for grouping of characters into tokens?

A. Parser  
B. Code optimization  
C. Code generator  
D. Lexical analyzer

20. Three address code involves

A. Exactly 3 address  
B. At most most 3 address  
C. No unary operators  
D. None of these

21. Shift reduce parsers are

A. Top down parser  
B. Bottom up parser  
C. May be top down or bottom up parser  
D. None of these

22. Whether a given pattern constitutes a token or not depends on the

A. Source language  
B. Target language  
C. Compiler  
D. All of these

23. Consider the languages $L_1$, $L_2$ and $L_3$ as given below.

$L_1 = \{0^q0^r | p, q \in \mathbb{N}\}$,

$L_2 = \{0^p1^q | p, q \in \mathbb{N} \text{ and } p = q\}$ and

$L_3 = \{0^p1^q0^r | p, q, r \in \mathbb{N} \text{ and } p = q = r\}$. Which of the following statements is NOT TRUE?

(A) Push Down Automata (PDA) can be used to recognize $L_1$ and $L_2$  
(B) $L_1$ is a regular language  
(C) All the three languages are context free  
(D) Turing machines can be used to recognize all the languages

24. In an absolute loading scheme, which loader function is accomplished by a loader?

A. Re-allocation  
B. Allocation  
C. Linking  
D. Loading
25. or scanning is the process where the stream of characters making up the source program is 
read from left to right and grouped into tokens.

A. Lexical analysis
B. Diversion
C. Modeling
D. None of the above

26. Compiler can diagnose

A. grammatical errors only
B. logical errors only
C. grammatical as well as logical errors
D. None of these

ANSWER : A

27. Find the near match between Group-1 and Group -2 item:

Group-1  Group -2
P. Data-flow analysis 1. Lexical analysis
Q: Regular expression 2. Semantic analysis
R: Type Checking 3. Parsing
S: Pushdown Automata 4. Code optimization
A. P-4, Q-1, R-3, S-2
B. P-2, Q-1, R-4, S-3
C. P-1, Q-4, R-2, S-3
Answer: ______________

28. Consider the grammar: S -> L= R | R
    L ->* R | id
    R -> L

Which of the following set of LR (0) items definitely does not represent a valid state 
of an LR (0) parser?
A. S -> L =·R, R -> ·L
B. L -> id.
C. S -> L· = R, R -> L·
D. R -> L·

29. Which of the following grammar rules violate the requirements of an operator grammar?
P,Q,R are non-terminals and r,s,t are terminals.
(i) P -> Q R
(ii) P -> s R
(iii) P -> ε
(iv) P -> Q t R r
(A) (i) only
(B) (i) and (iii) only
(C) (ii) and (iii) only
(D) (iii) and (iv) only

30. Which of the following statements is false?
   (A) An unambiguous grammar has same leftmost and rightmost derivation
   (B) An LL(1) parser is a top-down parser
   (C) LALR is more powerful than SLR
   (D) An ambiguous can never be LR(k) for any k

31. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G has n2 states. The relationship between n1 and n2 is
   (A) n1 is necessarily less than n2
   (B) n1 is necessarily equal to n2
   (C) n1 is necessarily greater than n2
   (D) None of these

32. Question: Consider the following two statements:
   S1: { 0^n 2^n | n >= 1} is a regular language
   S2: { 0^n 0^n 0^(m+n) | m >= 1 and n >= 2} is a regular language
Which of the following statements is correct?
   a) Only S1 is correct
   b) Only S2 is correct
   c) Both S1 and S2 are correct
   d) None of S1 and S2 is correct

33. Which of the following is used for grouping of characters into tokens (in a computer)?
   (A) A parser
   (B) Code optimizer
   (C) Code generator
   (D) Scanner

34. Which of the following is the most general phase structured grammar?
   (A) Regular
   (B) Context-sensitive
   (C) Context free
   (D) None of the above

35. For the grammar below, a partial LL(1) parsing table is also presented along with the grammar. Entires that need to be filled are indicated as $E_1, E_2$, and $E_3$. $\epsilon$ is the empty string, $\$ indicates end of input, and $|$ separates alternate right hand sides of productions.
   $$S \rightarrow aAbB \mid bAaB \mid \epsilon$$
The FIRST and FOLLOW set for the non-terminals A and B are

1. FIRST(A) = \{a,b,\epsilon\} = FIRST(B)
   FOLLOW(A) = \{a,b\}
   FOLLOW(B) = \{a,b,\$\}
2. FIRST(A) = \{a,b,\$\}
   FIRST(B) = \{a,b,\epsilon\}
   FOLLOW(A) = \{a,b\}
   FOLLOW(B) = \{\$\}
3. FIRST(A) = \{a,b,\epsilon\} = FIRST(B)
   FOLLOW(A) = \{a,b\}
   FOLLOW(B) = \emptyset
4. FIRST(A) = \{a,b,\epsilon\} = FIRST(B)
   FOLLOW(A) = \{a,b\}
   FOLLOW(B) = \{a,b\}

ANSWER__________

36. The appropriate entries for E1, E2 and E3 are

1. \textit{E1}: S \rightarrow aAbB, A \rightarrow S
   \textit{E2}: S \rightarrow bAaB, B \rightarrow S
   \textit{E3}: B \rightarrow S
2. \textit{E1}: S \rightarrow aAbB, S \rightarrow \epsilon
   \textit{E2}: S \rightarrow bAaB, S \rightarrow \epsilon
   \textit{E3}: S \rightarrow \epsilon
3. \textit{E1}: S \rightarrow aAbB, S \rightarrow \epsilon
   \textit{E2}: S \rightarrow bAaB, S \rightarrow \epsilon
   \textit{E3}: B \rightarrow S
4. \textit{E1}: A \rightarrow S, S \rightarrow \epsilon
   \textit{E2}: B \rightarrow S, S \rightarrow \epsilon
   \textit{E3}: B \rightarrow S
37. Consider the program given below, in a block-structured pseudo-language with lexical scoping and nesting of procedures permitted.

Program main;
Var ...
Procedure A1;
Var ...
Call A2;
End A1

Procedure A2;
Var ...

Procedure A21;
Var ...
Call A1;
End A21

Call A21;
End A2;

Call A1;
End main

Consider the calling chain: Main → A1 → A2 → A21 → A1
The correct set of activation records along with their access links is given by
A. B.
38. In a compiler, keywords of a language are recognized during
A. parsing of the program
B. the code generation
C. the lexical analysis of the program
D. dataflow analysis

39. Consider two binary operators '↑' and '↓' with the precedence of operator ↓ being lower than that of the operator ↑. Operator ↑ is right associative while operator ↓ is left associative. Which one of the following represents the parse tree for expression (7↓3↑4↑3↓2)?
A. 
B.
40. Which data structure in a compiler is used for managing information about variables and their attributes?
A. Abstract Syntax Tree  
B. Symbol Table  
C. Semantic Stack  
D. Parse Table

41. Which language necessarily need heap allocation in the runtime environment?
A. Those that support recursion  
B. Those that use dynamic scoping  
C. Those that allow dynamic data structures  
D. Those that use global variables

42. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G has n2 states. The relationship between n1 and n2 is
(A) n1 is necessarily less than n2  
(B) n1 is necessarily equal to n2
43. Which of the following is used for grouping of characters into tokens (in a computer)?
(A) A parser
(B) Code optimizer
(C) Code generator
(D) Scanner

ANSWER_____________

44. Which of the following is the most general phase structured grammar?
(A) Regular
(B) Context-sensitive
(C) Context free
(D) None of the above

ANSWER_____________

45. Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order.
(a) Leftmost derivation
(b) Leftmost derivation traced out in reverse
(c) Rightmost derivation
(d) Rightmost derivation traced out in reverse

ANSWER_____________

46. The lexical analysis for a modern language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?
(A) Finite state automata
(B) Deterministic pushdown automata
(C) Non-deterministic pushdown automata
(D) Turing machine

(C) n1 is necessarily greater than n2
(D) None of these

ANSWER_____________
47. In a bottom-up evaluation of a syntax directed definition, inherited attributes can
(A) always be evaluated
(B) be evaluated only if the definition is L-attributed
(C) be evaluated only if the definition has synthesized attributes
(D) never be evaluated

ANSWER________________

48. Which of the following derivations does a top-down parser and left to right order.
(a) Leftmost derivation
(b) Leftmost derivation in reverse
(c) Rightmost derivation
(d) Rightmost derivation in reverse

ANSWER________________

49. Consider the grammar
S -> A a | b
A -> A c | S d | ε
Construct an equivalent grammar with no left recursion and with minimum number
of production sales.

ANSWER________________

50. Which data structure in a compiler is used for managing information about variables
and their attributes?
(A) Abstract syntax tree (B) Symbol table
(C) Semantic stack (D) Parse table

ANSWER________________

51. Which one of the following regular expressions is NOT equivalent to the regular
expression (a + b + c) *?

ANSWER________________
52. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?
   (A) Removing left recursion alone
   (B) Factoring the grammar alone
   (C) Removing left recursion and factoring the grammar
   (D) None of the above

ANSWER___________

53. In a bottom-up evaluation of a syntax directed definition, inherited attributes can
   (A) always be evaluated
   (B) be evaluated only if the definition is L-attributed
   (C) be evaluated only if the definition has synthesized attributes
   (D) never be evaluated

ANSWER___________

54. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G
   has n2states. The relationship between n1 and n2 is
   (A) n1 is necessarily less than n2
   (B) n1 is necessarily greater than n2
   (C) n1 is necessarily equal to n2
   (D) None of the above

ANSWER___________

55. Which of the following derivations does a top-down parser use while parsing an input
   string? The input is assumed to be scanned in left to right order.
   (a) Leftmost derivation
   (b) Leftmost derivation traced out in reverse
   (c) Rightmost derivation
   (d) Rightmost derivation traced out in reverse

ANSWER___________

56. Consider the grammar:
   \[ S \rightarrow (S) | a \]
   Let the number of states in SLR (1), LR(1) and LALR(1) parsers for the grammar
be n1, n2 and n3 respectively. The following relationship holds good:
(a) n1 < n2 < n3
(b) n1 = n2 < n3
(c) n1 = n2 = n3
(d) n1 ≥ n2 ≥ n3

ANSWER__________

57. Consider the grammar:
   E -> E + n | E × n | n
For a sentence n + n × n, the handles in the right-sentential form of the reduction are:
(a) n, E + n and E + n × n
(b) n, E + n and E + E × n
(c) n, n + n and n + n × n
(d) n, E + n and E × n

ANSWER__________

58. Some code optimizations are carried out on the intermediate code because
(A) They enhance the portability of the compiler to other target processors
(B) Program analysis is more accurate on intermediate code than on machine code
(C) The information from dataflow analysis cannot otherwise be used for optimization
(D) The information from the front end cannot otherwise be used for optimization

ANSWER__________

59. Which of the following statements is false?
(a) An unambiguous grammar has same leftmost and rightmost derivation
(b) An LL(1) parser is a top-down parser
(c) LALR is more powerful than SLR
(d) An ambiguous grammar can never be LR(k) for any k

ANSWER__________

60. Consider the following translation scheme.
   S → ER
   R → *E {print('*'); R |ε
   E → F + E {print('+'); | F
   F → (S) | id {print (id.value);}
Here id is a taken that represents an integer and id.value represents the corresponding integer value. For an input ‘2 * 3 + 4’, this translation scheme prints
(A) $2 \times 3 + 4$
(B) $2 \times 3 4$
(C) $2 3 \times 4 +$
(D) $2 3 4 + *$

ANSWER________________

61. Consider the following grammar
   \[ S \rightarrow FR \]
   \[ R \rightarrow *S \mid \varepsilon \]
   \[ F \rightarrow \text{id} \]

   in the predictive parser table, \( M \), of the grammar the entries \( M[S, \text{id}] \) and \( M[R, \varepsilon] \) respectively
   (A) \{ \(S \rightarrow FR\) \} and \{ \(R \rightarrow \varepsilon\) \}
   (B) \{ \(S \rightarrow FR\) \} and \{ \}
   (C) \{ \(S \rightarrow FR\) \} and \{ \(R \rightarrow *S\) \}
   (D) \{ \(F \rightarrow \text{id}\) \} and \{ \(R \rightarrow \varepsilon\) \}

ANSWER________________

62. Consider the grammar shown below: \( S \rightarrow C C \)
   \( C \rightarrow c C \mid d \)

   The grammar is
   (A) LL (1)
   (B) SLR (1) but not LL (1)
   (C) LALR (1) but not SLR (1)
   (D) LR (1) but not LALR (1)

ANSWER________________

63. Consider the grammar \( S \rightarrow (S) \mid a \)

   Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar be \( n_1 \) \( n_2 \) and \( n_3 \) respectively. The following relationship holds good
   (A) \( n_1 < n_2 < n_3 \)
   (B) \( n_1 = n_3 < n_2 \)
   (C) \( n_1 = n_2 = n_3 \)
   (D) \( n_1 \geq n_3 \geq n_2 \)

ANSWER________________

64. Which of the following is the most powerful parsing method?
   (A) LL (1)
   (B) Canonical LR
   (C) SLR
(D) LALR

ANSWER__________

65. Context-free languages are closed under:
   (A) Union, intersection
   (B) Union, Kleene closure
   (C) Intersection, complement
   (D) Complement, Kleene Closure

ANSWER__________

66. The number of tokens in the Fortran statement DO 10 I = 1.25 is
   (A) 3
   (B) 4
   (C) 5
   (D) None of these

ANSWER__________

67. The grammar $S \rightarrow aSa \mid bS \mid c$ is
   (A) LL(1) but not LR(1)
   (B) LR(1) but not LL(1)
   (C) Both LL(1) and LR(1)
   (D) Neither LL(1) nor LR(1)

ANSWER__________

68. Let $L_1$ be a recursive language. Let $L_2$ and $L_3$ be languages that are recursively
   enumerable but not recursive. Which of the following statements is not
   necessarily true?
   (A) $L_2 \setminus L_1$ is recursively enumerable
   (B) $L_1 \setminus L_3$ is recursively enumerable
   (C) $L_2 \cap L_1$ is recursively enumerable
   (D) $L_2 \cup L_1$ is recursively enumerable

ANSWER__________

69. Which data structure in a compiler is used for managing information about
   variables and their attributes?
   (A) Abstract syntax tree (B) Symbol table
   (C) Semantic stack (D) Parse table

ANSWER__________
70. A grammar that is both left and right recursive for a non-terminal, is 
(a) Ambiguous (b) Unambiguous 
(c) Information is not sufficient to decide whether it is ambiguous or unambiguous 
(d) None of the above

ANSWER___________

71. Which of the following is the most powerful parsing method? 
(a) LL (1) (b) Canonical LR (c) SLR (d) LALR

ANSWER___________

72. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar? 
(A) Removing left recursion alone 
(B) Factoring the grammar alone 
(C) Removing left recursion and factoring the grammar 
(D) None of the above

ANSWER___________

73. Given the following expression grammar: 
   
   \[
   E \rightarrow E \times F \mid F+E \mid F \\
   F \rightarrow F-F \mid F \mid id \\
   \]

   which of the following is true? 
   (a) \( \times \) has higher precedence than \( + \) 
   (b) \( - \) has higher precedence than \( \times \) 
   (c) \( + \) and \( - \) have same precedence 
   (d) \( + \) has higher precedence than \( \times \)

ANSWER___________

74. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar? 
(a) Removing left recursion alone 
(b) Factoring the grammar alone 
(c) Removing left recursion and factoring the grammar 
(d) None of the above

ANSWER___________

75. In a compiler, keywords of a language are recognized during 
(A) parsing of the program (B) the code generation 
(C) the lexical analysis of the program (D) dataflow analysis


ANSWER___________

76. Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order
(a) Leftmost derivation (b) Leftmost derivation traced out in reverse
(c) Rightmost derivation (d) Rightmost derivation traced out in reverse
ANSWER___________

77. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?
(a) Removing left recursion alone (b) Factoring the grammar alone
(c) Removing left recursion and factoring the grammar (d) None of the above

78. Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order (GATE CS 2000).
(a) Leftmost derivation
(b) Leftmost derivation traced out in reverse
(c) Rightmost derivation
(d) Rightmost derivation traced out in reverse
Answer (a)

79. The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called (GATE CS 2001)
(a) Assembly 
(b) Parsing 
(c) Relocation 
(d) Symbol resolution 
Answer: (c)

80. Which of the following statements is false? (GATE CS 2001)
(a) An unambiguous grammar has same leftmost and rightmost derivation
(b) An LL(1) parser is a top-down parser
(c) LALR is more powerful than SLR
(d) An ambiguous grammar can never be LR(k) for any k
Answer: (a)

81. Which of the following grammar rules violate the requirements of an operator grammar? P, Q, R are nonterminals, and r,s,t are terminals (GATE CS 2004).
(i) P -> QR
(ii) P -> QsR
(iii) P ->
(iv) P -> QtRr
(a) (i) only
(b) (i) and (iii) only
(c) (ii) and (iii) only
d) (iii) and (iv) only
Answer: (b)

82. Consider the grammar with the following translation rules and E as the start symbol.

\[
E \rightarrow E1 \#T \{E.value = E1.value \times T.value\} \\
| T \{E.value = T.value\} \\
T \rightarrow T1 \& F \{T.value = T1.value + F.value\} \\
| F \{T.value = F.value\} \\
F \rightarrow \text{num} \{F.value = \text{num.value}\}
\]

Compute E.value for the root of the parse tree for the expression: 2 \# 3 \& 5 \# 6 \& 4. (GATE CS 2004)

a) 200  
b) 180  
c) 160  
d) 40
Answer: (c)

83. Given the following expression grammar:

\[
E \rightarrow E \ast F | F+E | F \\
F \rightarrow F-F | id
\]

which of the following is true? (GATE CS 2000)

(a) \ast has higher precedence than +  
(b) – has higher precedence than *  
(c) + and — have same precedence  
(d) + has higher precedence than *
Answer (b)

84. Consider a program P that consists of two source modules M1 and M2 contained in two different files. If M1 contains a reference to a function defined in M2 the reference will be resolved at (GATE CS 2004)

a) Edit time  
b) Compile time  
c) Link time  
d) Load time
Answer (c)

85. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar? (GATE CS 2003)

(a) Removing left recursion alone  
(b) Factoring the grammar alone  
(c) Removing left recursion and factoring the grammar  
(d) None of the above
Answer (d)

86. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for
G has n2 states. The relationship between n1 and n2 is (GATE CS 2003)
(a) n1 is necessarily less than n2
(b) n1 is necessarily equal to n2
(c) n1 is necessarily greater than n2
(d) none of the above
Answer (b)