MADE EASY PUBLICATIONS

POSTAL BOOK PACKAGE 2025

CONTENTS

COMPUTER SCIENCE & IT

Objective Practice Sets

Compiler Design

Note: This book contains copyright subject matter to MADE EASY Publications, New Delhi. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means. Violators are liable to be legally prosecuted.

1 CHAPTER

Introduction to Compiler

Multiple Choice Questions

Q.1 Which translator program converts assembly language program to object program?
(a) Assembler (b) Compiler
(c) Microprocessor (d) Linker

Q.2 Which one is a phase of a compilation process?
(a) Lexical analysis (b) Code generation
(c) Both (a) and (b) (d) None of these

Q.3 Choose the correct sequence of occurrence during compilation process.
(a) Character stream → Parse Tree → Optimized code
(b) Parse tree → Token stream → Intermediate code
(c) SDT tree → Parse tree → Optimized code
(d) Parse tree → 3 address code → Character stream

Q.4 Assembly language
(a) is usually the primary user interface
(b) requires fixed format commands
(c) is a mnemonic form of machine language
(d) is quite different from the SCL interpreter

Q.5 For which of the following reasons compiler is preferable to an interpreter?
(a) It can generate stand-alone programs that often take less time for execution.
(b) It is much helpful in the initial stages of program development.
(c) Debugging can be faster and easier
(d) It needs less computer resources

Q.6 In a context-free grammar
(a) ε can't be the right-hand side of any production
(b) terminal symbols can't be present i the left-hand side of any production.
(c) the number of grammar symbols in the left-hand side is not greater than the number of grammar symbols in the right hand side.
(d) All of the above

Q.7 The cost of developing a compiler is proportional to the
(a) complexity of the source language
(b) complexity of the architecture of the target machine
(c) flexibility of the available instruction set
(d) All of the above

Q.8 Semantic errors can be detected by the system
(a) At compile time only
(b) At run time only
(c) Both at compile and run time
(d) None of these

Q.9 Undeclared name is _______ error.
(a) Syntax (b) Lexical
(c) Semantic (d) No

Q.10 A simple two-pass assembler does not do 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) It generates code for all the load and stores register instructions.

Q.11 Consider the following statement:
S_1 : Analysis phase of compiler includes code optimization stages.
S_2 : Synthesis phase of compiler is followed by analysis phase.
(a) S_1 is correct, S_2 is not
(b) S_2 is correct, S_1 is not
(c) S_1 and S_2 are correct
(d) S_1 and S_2 are incorrect

Q.12 Match the following groups:

List-I
A. Lexical analyzer
B. Syntax analyzer
C. Type checking
D. Intermediate code generation

List-II

1. Checks the structure of the program.
2. Analysis of entire program by reading each character.
3. High level language is translated to simple machine independent language.
4. Checks the consistency requirements in a context of the program.

Codes:

	A	B	C	D
(a)	1	2	4	3
(b)	2	1	4	3
(c)	2	4	3	1
(d)	1	4	3	2

Q.13 An optimizing compiler
(a) is optimized to occupy less space
(b) is optimized to take less time for execution
(c) optimized the code
(d) None of the above

Q.14 Which of the following is not a functionality of C compiler?
(a) Identifying syntax error
(b) Identifying tokens
(c) Linking
(d) None of these

Q.15 Which of the following grammars are not phase structured?
(a) Regular (b) Context-free
(c) Context-sensitive (d) None of the above

Q.16 Cross-compiler is a compiler
(a) which is written is a language that a different from the source language.
(b) that generates object code for its last machine.
(c) which is written in a language that is same as the source language.
(d) that runs on one machine but produces object code for another machine.

Q.17 Match **List-I** with **List-II** and select the correct answer using the code given below:

List-I	List-II
A. Load time	1. Relocation
B. Compile time	2. Token recognition
C. Link time	3. Resolving reference
D. Run time	4. Activation record

Code:

	A	B	C	D
(a)	1	4	3	2
(b)	1	2	3	4
(c)	1	3	2	4
(d)	4	1	3	2

Q.18 If w is a string of terminal and A, B are two non-terminals, then which of the following are right-linear are grammars?
(a) $A \rightarrow B\ W$
(b) $A \rightarrow B\ W/W$
(c) $A \rightarrow WB/W$
(d) None of the above

Q.19 CSG can be recognized by
(a) pushdown automata
(b) 2-way linear bounded automata
(c) finite state-automata
(d) None of the above

Q.20 A Top-down parser generates
(a) left-most derivation
(b) right-most derivation
(c) right-most derivation in reverse
(d) left-most derivation in reverse

Q.21 A programming language is to be designed to run on a machine that does not have a big memory. The language should
(a) prefer a 2 pass compiler to a 1 pass compiler
(b) prefer an interpreter to a compiler
(c) not support recursion
(d) All of the above

Q.22 A loader is
(a) a program that place programs into memory and prepares them for execution
(b) a program that automates the translations of assembly language into machine language
(c) a program that accepts a program written in a high level language and produces an object program
(d) a program that appears to execute a source program as if it were machine language

Q.23 Which of the following is the most general phase-structured grammar?
(a) Regular (b) Context-free
(c) Context-sensitive (d) None of the above

Q.24 An ideal compiler should
(a) be smaller in size
(b) be written in a high level language
(c) produce object code that is smaller in size and executes faster
(d) All of the above

Q.25 Whether a given pattern constitutes a token or not?
(a) depends on the source language
(b) depends on the target language
(c) depends on the compiler
(d) None of the above comment is true

Q.26 In a compiler, grouping of characters into tokens is done by the
(a) Scanner (b) Parser
(c) Code generator (d) Code optimizer

Q.27 Which of following is used for grouping of characters into tokens (in a compiler)
(a) Parser (b) Code optimizer
(c) Code generator (d) Scanner

Q.28 Consider the following C fragment

fro (int $x = 0$; $x <= n$; x++)

Which type of error detected by the C compiler for the above code?
(a) Lexical error (b) Syntactic error
(c) Semantic error (d) Logical error

Q.29 A grammar will be meaningless if the
(a) terminal set and non-terminal set are not disjoint.
(b) left hand side of a productions is a single terminal
(c) left hand side of a production has no none terminal
(d) All of the above

Q.30 Which of the following is correct?
(a) Loader resolves external memory references, when the code is one file may refer to a location in another file
(b) It is easy to design compiler for different source language and target machines because of the two phase division of compiler.
(c) The storage used for heap section can grow at run time but not stack section
(d) None of these

Q.31 Match **List-I** and **List-II** and select the correct answer using codes given below:

List-I
A. Expanding macros into source language statements.
B. It takes the output generated by the compiler as input and generate relocatable machine code as the output.
C. Puts together all the executable object files into memory for execution.

List-II
1. Assembler
2. Loader
3. Preprocessor

Code:

	A	B	C
(a)	3	2	1
(b)	3	1	2
(c)	1	2	3
(d)	1	3	2

Q.32 Incremental-compiler is a computer
(a) which is written in a language that is different from the source language.
(b) that generates object code for its host machine.
(c) which is written in a language that is same as the source language
(d) that allows a modified position of a programme to be compiled

Q.33 In a context-sensitive grammar.
(a) ε can't be the right-hand side of any production
(b) number of grammar symbols on the left-hand side of a production can't be greater than the number of non-terminals on the right hand side.
(c) number of grammar symbols on the left-hand side of a production can't be greater than the number of grammar symbols on the right-hand side.
(d) All of the above

Q.34 If a is a terminal and S, A, B are three non-terminals, then which of the following are regular grammar,
(a) $S \to \in$, $A \to as \mid B$
(b) $A \to aA \mid a$, $B \to bB \mid a$
(c) $A \to Ba \mid BaB$
(d) $A \to aBB \mid aB$

Q.35 Choose the correct statements:
(a) Sentence of a grammar is a sequential form without any terminals.
(b) Sentence of a grammar should be derivable from the start state.
(c) Sentence of a grammar should be frontier of a derivation tree, in which the root node has the start state as the label.
(d) None of the above

Q.36 Representing the syntax by a grammar is advantageous because
(a) It is concise
(b) It is accurate
(c) Automation becomes easy
(d) All of the above

Q.37 CFG can be recognized by a
(a) push down automata
(b) finite state automata
(c) 2-way linear bounded automata
(d) Non finite state automata

Multiple Select Questions (MSQ)

Q.38 For which of the following reasons, an interpreter is preferred to a compiler?
(a) It takes less time to execute
(b) It is much helpful in the initial stages of program development.
(c) Debugging is easier.
(d) It needs less computer resources.

Q.39 Storage mapping is done by
(a) Operating system (b) Compiler
(c) Linker (d) Loader

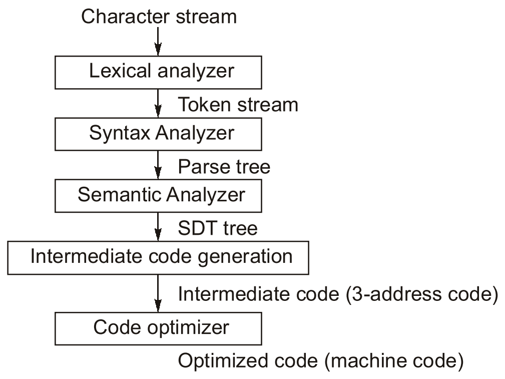
■■■■

Answers Introduction to Compiler

1.	(a)	2.	(c)	3.	(a)	4.	(c)	5.	(a)	6.	(b)	7.	(d)	8.	(a)	9.	(c)
10.	(d)	11.	(d)	12.	(b)	13.	(c)	14.	(c)	15.	(d)	16.	(d)	17.	(b)	18.	(c)
19.	(b)	20.	(a)	21.	(d)	22.	(a)	23.	(c)	24.	(d)	25.	(c)	26.	(a)	27.	(d)
28.	(b)	29.	(b)	30.	(b)	31.	(b)	32.	(d)	33.	(d)	34.	(b)	35.	(b)	36.	(d)
37.	(a)	38.	(b, c)	39.	(a, d)												

Explanations Introduction to Compiler

3. (a)



9. (c)

It is a semantic error (run-time error).

11. (d)

Analysis phase {lexical analysis, syntax analysis, semantic analysis} is followed by synthesis phase {intermediate code generation, code optimizer, machine code generation}

12. (b)

Lexical analyzer: Reads every character of the program to identify the tokens.
Syntax analyzer: Analyzes the syntax or structure of the program.
Type checking: It determines violation of consistency requirements.
ICG: Translates the program into intermediate language.

14. (c)

Linking is done by a linker after compilation process.
Compilation can identify token generates compilation error which can be lexical, syntax or semantic.

17. (b)

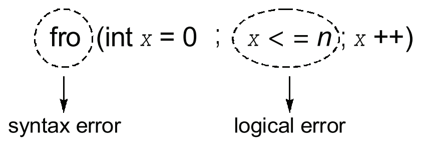
Load time relocation is done on load time:
Compile time : Token recognition during complication
Line time : Reference can be resolved
Run time : Activation record created during run time

21. (d)

Since the programming language is to be designed to run on a machine that does not have a big memory. Therefore pass-2 (multi phase), which uses less space is preferred also instead of using a compiler we use an interpreter, which scans the program line by line hence uses less memory at compiler time for each instance. Also it does not support recursion.

28. (b)

Although the statement



Has both syntax and logical error
But, it will result in syntax error.
Hence, (b) is correct option.

30. (b)

The compiler is divided into two phase all the compiler modules from lexical analysis till ICG are in front end and alter that back end.

31. (b)

The preprocessor expand macros, into source language statements and generate modified source program as the output.
The assembly language is processed by a program called an assembles that produces relocatable machine code.
Loader puts together all of the executable object files into memory for execution.

■■■■

2 CHAPTER Lexical Analysis

Multiple Choice Questions & NAT Questions

Q.1 The task of the lexical analysis phase is
(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 the above

Q.2 The output of lexical analyzer is
(a) a set of regular expressions
(b) syntax tree
(c) set of tokens
(d) strings of characters

Q.3 How many tokens are contained in the following FORTRAN statement:

IF(NUMB EQ MAX) GOTO 500

(a) 8 (b) 10
(c) 22 (d) 24

Q.4 The number of tokens in the following C statement

printf("i = %d, & i = % x", i, & i);

(a) 3 (b) 26
(c) 10 (d) 21

Q.5 Consider the following statements:
1. Lexical analysis removes white spaces and not the comments.
2. for (i = 0) generates lexical error in C
3. Lexeme is the string and not the pattern.
4. fi (a > b) generates no lexical error in C.

Identify the false statements:
(a) Only 1 and 4 (b) Only 1 and 2
(c) Only 2 (d) Only 4

Q.6 Consider the following C-program:

```
int strange (int x)
{
  if (x < = 0) return 0;
  if (x% 2! = 0) return x – 1;
  return 1 + strange (x – 1);
}
```

Find out the number of tokens?
(a) 44 (b) 42
(c) 43 (d) 75

Q.7 Consider line-3 of the following c-program:

```
int main ( ) {          /* line1*/
int i, n ;              /* line2*/
For (i = 0; i < n; i ++); /* line 3*/
```

Identify the compiler response about the line 3 while creating the object module.
(a) No compilation error
(b) Only a lexical error
(c) Only Syntactic error
(d) Both lexical error and syntactic error

Q.8 Find the line number in which lexical error present in the following program:

```
1. main( )
2. {
3.      int x; y; z;
4.      /* comment1*/
5.      /*com/*ment2*/end*/
6.      x = "A";
7. }
```

(a) 3 (b) 5
(c) 6 (d) No lexical error

Q.9 Consider the following program:

```
main( )
{
    char ch = 'A';
    int x, y;
    x = y = 20;
    x ++;
    printf("%d% d", x, y);
}
```

The number of tokens in the above program are ________.

Q.10 Which of the following equivalent one is used in lexer?

(a) Regular expression
(b) Context free language
(c) Both (a) and (b)
(d) Neither (a) nor (b)

Q.11 In some programming languages, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denotes the sets of letters and digits respectively, which expression defines an identifier?
(a) $(LUD)^+$ (b) $L(LUD)^*$
(c) $(L.D)^*$ (d) $L.(L.D)^*$

Q.12 Consider the following C-program. Find the number of tokens in the output of the following program if that is passed as a string to a lexical analyzer.

```
main( )
{
    char ∗ s[ ] = {“ice”, “green”, “water”, “hi”};
    char ∗∗ ptr[ ] = {s + 3, S + 2, S + 1, s}
    char ∗∗∗ p = ptr;
printf(“%s”, ∗∗ ++ P);
}
```

(a) 5 (b) 1
(c) 4 (d) 3

Q.13 Find the number of tokens in the following C code using lexical analyzer of compiler.

```
main( )
{
   int ∗a, b;
   b = 10;
   a = &b;
   printf(“%d%d”, b, *a);
   b = ∗/∗pointer∗/b;
}
```

(a) 35 (b) 45
(c) 55 (d) 65

Q.14 Count number of tokens in the following given code:

```
main( )
{
    int a;
    int* a;
    For (i = 0; i < n; i++)
    {
      printf(“True”);
      ++* a;
      a = a + a;
    }
}
```

Q.15 Find the regular expression that correctly identifies the variable name in C program.
(a) [a – z][a – zA – Z_O – 9]*
(b) [a – zA – Z_][a – zA – Z_O – 9]*
(c) [a – zA – Z_][a – zA – ZO – 9]*
(d) [a – zA – Z][a – zA – Z_O – 9]*

Q.16 Consider the following code:

```
int a = 10;
Float b = 20.5;
Printf(“c = %d, d = %”, ++a, b++);
```

The number of tokens in the given code fragment is ________.

Q.17 Consider the following C Program:

```
Void main( )
{
    int i = 20;
    while(i – –)
    Printf(“GATE 2020\n”);
}
```

The output of the given program is
(a) Prints Gate 2020 20 Times
(b) Prints Gate 2020 19 times
(c) Lexical error
(d) Syntax error

Q.18 Which of the following errors that are detected in the Lexical Analysis Phase.
1. Numeric literals that are too long.
2. Identifiers that are too long.
3. Ill-formed numeric literals.
4. Input characters that are not in language.

(a) 1 and 2 only
(b) 3 and 4 only
(c) 1, 3 and 4 only
(d) 1, 2, 3 and 4

Q.19 Consider the following CFG:

$$S \rightarrow AaC \mid aA$$
$$A \rightarrow bC \mid C$$
$$C \rightarrow aC \mid a$$

Which of the following is true about the above grammar?
(a) Left recursive and unambiguous
(b) Non left recursive and ambiguous
(c) Ambiguous and left recursive
(d) None of these

Answers Lexical Analysis

1.	(a)	2.	(c)	3.	(a)	4.	(c)	5.	(b)	6.	(b)	7.	(c)	8.	(d)	9.	(33)
10.	(a)	11.	(b)	12.	(b)	13.	(a)	14.	(42)	15.	(b)	16.	(21)	17.	(d)	18.	(d)
19.	(b)	20.	(18)	21.	(46)	22.	(c)	23.	(41)	24.	(24)	25.	(a, b, c, d)			26.	(a, b)

Explanations Lexical Analysis

3. (a)

The given statement can be broken into

$\underline{\text{IF}}\ \underline{(}\ \underline{\text{NUMB}}\ \underline{\text{EQ}}\ \underline{\text{MAX}}\ \underline{)}\ \underline{\text{GOTO}}\ \underline{\text{50C}}$

∴ Number of tokens = 8

4. (c)

Number of tokens in C statement is 10

$\underline{\text{Printf}}\ \underline{(}\ \underline{\text{"I = \%d, \& I=\%x}}\ \underline{,}\ \underline{\text{i}}\ \underline{,}\ \underline{\&}\ \underline{\text{I}}\ \underline{)}\ \underline{;}$

5. (b)

Lexical analysis removes comments as well.
(i = 0) will not generate lexical error. It will tokenise as for, (, i, =, 0,). No error is generated.

6. (b)

```
 ①      ②       ③  ④ ⑤ ⑥
|int| strange| ( |int|x|  )|
 ⑦
|{|

 ⑧  ⑨ ⑩ ⑪  ⑫ ⑬   ⑭     ⑮ ⑯
|if| ( |x| <=| 0| ) | return| 0 | ;|

⑰ ⑱ ⑲ ⑳ ㉑ ㉒㉓  ㉔ ㉕ ㉖   ㉗      ㉘ ㉙ ㉚ ㉛
|if| ( | ( |x |%| 2| ) |!=| 0| )| return| x |–| 1| ;|
  ㉜      ㉝ ㉞    ㉟       ㊱ ㊲ ㊳ ㊴ ㊵ ㊶
|return| 1 | + |strange| ( |x | – | 1| ) | ;|
㊷
|}|
```

7. (c)

The C-code contains syntax error, where 'fro' is written in place of 'for'. The lexical analyzer will not declare it as an error, whether 'fro' is a misspelling of the keyword 'for' or an undeclared function identifier. Since 'fro' is a valid lexeme for the token id, the lexeme analyzer must return the token id to the parser and let some other phase of the compiler (parser in this case) handle the error due to transposition of the letters.

8. (d)

The given program contain no lexical error even though it contains syntax errors. In line number "5", comment started and searches for the first close comment pattern when it finds, it consider a comment. There is no start comment pattern (/*) but there is end comment at last in line 5, hence it is not lexical error but it is syntax error.

$\underbrace{\text{/ * com / * ment2 * /}}_{\text{Comment}}\ \underbrace{\text{end}}_{\text{Identifier}}\ \underbrace{\text{* /}}_{\text{Operator}}$ ⇒ No lexical error

9. (33)

```
main  (  )
 ①   ②  ③
{
④
   char ch  =  'A'  ;
    ⑤   ⑥  ⑦  ⑧  ⑨
   int x  ,  y  ;
   ⑩  ⑪ ⑫ ⑬ ⑭
   x  =  y  =  20  ;
   ⑮ ⑯ ⑰ ⑱ ⑲ ⑳
   x  ++  ;
   ㉑ ㉒ ㉓
   printf  (  "%d%d"  ,  x  ,  y  )  ;
     ㉔   ㉕    ㉖    ㉗ ㉘ ㉙ ㉚ ㉛ ㉜
}
㉝
```

10. (a)

Regular expression is used in lexical analysis to identify the tokens.

12. (b)

We need not to find out the actual output of the given C-program. Just by seeing the C-code we can understand that the output is a string and that is passed to lexical analyzer and that would be treated as a single valid lexeme for lexical analyzer. For instance we can see the output of the program will be 'water'.

$\left|\underset{(1)}{\text{Water}}\right|$ is considered as a single token.

13. (a)

```
main ( )
{
   int * a , b ;
   b = 10 ;
   a = & b ;
   printf ( "%d%d" , b , * a ) ;
   b = * /*pointer*/ b ;
}
```

14. (42)

Code	Number of Tokens
Main ()	3
{	1
int a;	3
int*a;	4
For (i = 0; i < n; i ++)	13
{	1
Printf ("True")	5
++* a;	4
a = a + a;	6
}	1
}	1
	Total = 42

15. (b)

In C program, Variable Name consists of letters, digits and underscore symbol. Variable name must begin with a letter or underscore.

[a – zA – Z_][a – zA – Z_O – 9]*

16. (21)

int a = 10; Number of tokens are 5.

Float b = 20.5; Number of tokens are 5.

Printf("c = %d, d = %F", ++a, b++);

Number of tokens are 11

Total number of tokens are = 21

17. (d)

Compiler produces an error as the spelling of the keyword while is written as while. Lexical analyzer treats it is a valid identifier. Syntax analyzer will found and display an error message.

18. (d)

Lexical Analysis will detect all the mentioned errors. Numeric literals that are too long.

Identifiers that are too long.

ICC-formed numeric literals.

Input characters that are not in the language.

19. (b)

$S \to AaC \,/aA$

$A \to bC/C$

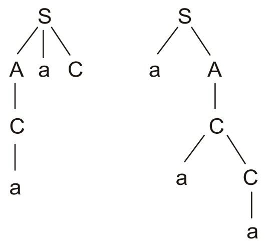
$C \to aC/a$

Grammar is non left recursive

$A \to bC\,/C$

$C \to aCa$

aaa can be generated by two different Parse tree.



So, it is ambiguous.

20. (18)

$S \to aA\ BbCD$

$A \to A\ Sd\ |\varepsilon$

$B \to SAc\ |hC|\varepsilon$

$C \to Sf|Cg$

$D \to BD|\varepsilon$

Delete $D \to \varepsilon$

$S \to aA\ BbCD|aA\ Bb\ C$

$A \to A\ Sd|\varepsilon$

$B \to SAc|hC|\varepsilon$

$C \to Sf|Cg$

$D \to BD|B$

Delete $B \to \varepsilon$

$S \to aA\ BbCD|aABbC|aAbCD|aAbC$

$A \to A\ Sd\ |\varepsilon$

$B \to SAc|hC$

$C \to Sf|Cg$

$D \to BD|B|D$

Delete $A \to \varepsilon$

$S \to aA\ BbCD|aABbC|aAbCD|aAbC|$
$aBbCD|aB\ bC|abCD|abC$

$A \to A\ Sd|Sd$

$B \to SAc|hC|Sc$

$C \to Sf|Cg$

$D \to BD|B|D$

18 productions are present after removal of all null productions.

3 CHAPTER Syntax Analysis (Parser)

Multiple Choice Questions & NAT Questions

Q.1 Which of the following suffices to convert an unambiguous 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 above

Q.2 Non backtracking form of top-down parser are called
(a) recursive-descent parsers
(b) predictive parsers
(c) shift reduce parsers
(d) None of these

Q.3 Parsing table in the LR parsing contains
(a) action, goto (b) state, action
(c) input, action (d) state, goto

Q.4 Consider the following grammar
expr $\rightarrow$ expr op expr
expr $\rightarrow$ id
op $\rightarrow$ + | *
Which of the following is true?
(a) op and expr are start symbols
(b) op and id are terminals
(c) expr is start symbol and op is non terminal
(d) none of these

Q.5 Consider the grammar:
$S \rightarrow aABe$
$B \rightarrow d$
$A \rightarrow Abc \mid b$
For which the string "*abbcde*" is right most derivation
(a) $S \rightarrow aABe \Rightarrow aAde \Rightarrow aAbcde \Rightarrow abbcde$
(b) $S \Rightarrow aABe \Rightarrow aAbcBe \Rightarrow abbcBe \Rightarrow abbcde$
(c) Both (*a*) and (*b*)
(d) None of these

Q.6 Suppose a production $A \rightarrow \alpha B$ or $A \rightarrow \alpha B\beta$ where FIRST (β) contains $\in$ then which of the following is true?
(a) FOLLOW A is in FOLLOW B
(b) FOLLOW B is in FOLLOW A
(c) $\in$ will be in FOLLOW A
(d) Both (a) and (b)

Q.7 Choose the false statements.
(a) LL(*k*) grammar has to be a CFG
(b) LL(*k*) grammar has to be unambiguous
(c) There are LL(*k*) grammars that are not context free
(d) LL(*k*) grammars cannot have left recursive non-terminals

Q.8 The operator in a grammar, which is derived first, has _______ precedence.
(a) High
(b) Low
(c) Depends on grammar
(d) Cannot say

Q.9 For an arbitrary grammar, number of states in LALR(1) parser is ______ number of states in LR(1) parser
(a) $\geq$ (b) $\leq$
(c) $=$ (d) $\neq$

Q.10 The grammar is
$S \rightarrow A \mid B$
$A \rightarrow a \mid b$
$B \rightarrow b \mid c$
(a) LL(1) (b) LR(1)
(c) Both (a) and (b) (d) None of these

Q.11 Backtracking is possible in
(a) LR parsing
(b) Predictive parsing
(c) Recursive descent parsing
(d) None of the above

Q.12 Which of the following statements is true?
(a) SLR parser is more powerful than LALR.
(b) LALR parser is more powerful than canonical LR parser.

(c) LR parser is more powerful than LALR parser.
(d) SLR, LR and LALR parsers have the same power.

Q.13 In which of the following has attribute values at each node?
(a) Abstract syntax tree
(b) Postfix parse tree
(c) Annotated parse tree
(d) Prefix parse tree

Q.14 Operator-precedence parsing method is a parsing method. Which of the following statement is false about it?
1. It is bottom-up parsing method.
2. It must contain $\in$-production.
3. It doesn't contain two adjacent non-terminal symbols.

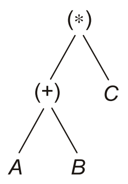
(a) 1 only (b) 2 only
(c) 3 only (d) 1 and 3 only

Q.15 Which of the following derivations does a top-down parser use while parsing as input string? The input is assumed to be scanned in left to right order
(a) Left most derivation
(b) Left most derivation traced out in reverse
(c) Right most derivation
(d) Right most derivation traced out in reverse

Q.16 A bottom-up parser generates
(a) right-most derivation
(b) right-most derivation in reverse
(c) left-most derivation
(c) left-most derivation in reverse

Q.17 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

Q.18 Which of the following expressions is represented by the parse tree?



(a) $(A + B) * C$ (b) $A + * BC$
(c) $A + B * C$ (d) $A * C + B$

Q.19 Which of the following parsers is the most powerful?
(a) Operator-precedence (b) Canonical LR
(c) LALR (d) SLR

Q.20 Handle pruning is the technique used to obtain
(a) Canonical derivation sequence
(b) Canonical reduction sequence
(c) Both (a) and (b) above
(d) None of the above

Q.21 Consider the following left-associative operators, in decreasing order of precedence:
– subtraction (highest precedence)
$*$ multiplication
\$ exponentiation (lowest precedence)

What is the result of the following expression?
3 – 2 * 4 \$ 2 \$ 3
(a) – 61 (b) 64
(c) 512 (d) 4096

Q.22 Consider the following grammar G:
$E \rightarrow TF$
$T \rightarrow xT$
$F \rightarrow y$
The number of states in LALR parser for G is ________.

Q.23 Which of the following actions an operator-precedence parser may take to recover from an error?
(a) Insert symbols onto the stack
(b) Delete symbols from the stack
(c) Insert or delete symbols from the input
(d) All of the above

Q.24 How many conflicts are there in the following grammar
$$S \rightarrow SS \mid a \mid \in$$
(a) 3 (b) 4
(c) 5 (d) 6

Direction: Q.25 and Q.26:
$S \rightarrow aSAb \mid bSBC$
$A \rightarrow +AB \mid \in$
$B \rightarrow BC \mid \in$
$C \rightarrow aC \mid d$

Q.25 What is in the follow (S)?
(a) {a, b, c, +, \$} (b) {a, c, +, $*$, \$}
(c) {b, c, +, $*$, \$} (d) {a, b, d, $*$, \$}

Answers Syntax Analysis (Parser)

1.	(c)	2.	(b)	3.	(a)	4.	(c)	5.	(a)	6.	(a)	7.	(c)	8.	(b)	9.	(b)
10.	(d)	11.	(c)	12.	(c)	13.	(c)	14.	(b)	15.	(a)	16.	(b)	17.	(c)	18.	(a)
19.	(b)	20.	(b)	21.	(d)	22.	(7)	23.	(d)	24.	(b)	25.	(c)	26.	(a)	27.	(b)
28.	(b)	29.	(c)	30.	(c)	31.	(b)	32.	(a)	33.	(a)	34.	(d)	35.	(c)	36.	(a)
37.	(c)	38.	(d)	39.	(c)	40.	(1)	41.	(d)	42.	(10)	43.	(c)	44.	(c)	45.	(c)
46.	(0)	47.	(c)	48.	(a)	49.	(a)	50.	(b)	51.	(a)	52.	(b)	53.	(b)	54.	(b)
55.	(a)	56	(a)	57.	(a)	58.	(a)	59.	(a)	60.	(c)	61.	(d)	62.	(b)	63.	(b)
64.	(d)	65.	(b)	66.	(d)	67.	(c)	68.	(2)	69.	(c)	70.	(23131)	71.	(c)		
72.	(d)	73.	(a)	74.	(a)	75.	(b)	76.	(7)	77.	(c)	78.	(b)	79.	(d)	80.	(c)
81.	(a)	82.	(c)	83.	(d)	84.	(b)	85.	(a)	86.	(c)	87.	(c)	88.	(7)	89.	(d)
90.	(b)	91.	(c)	92.	(c)	93.	(d)	94.	(b)	95.	(a)	96.	(d)	97.	(d)	98.	(d)
99.	(b)	100.	(8)	101.	(a)	102.	(b)	103.	(a)	104.	(3)	105.	(1)	106.	(9)	107.	(a)
108.	(7)	109.	(c)	110.	(a)	111.	(b)	112.	(5)	113.	(d)	114.	(d)	115.	(d)	116.	(d)
117.	(c)	118.	(c)	119.	(c)	120.	(c)	121.	(a)	122.	(c)	123.	(a)	124.	(b, c)	125.	(b, c)
126.	(b, c)	127.	(b, c)	128.	(a, b, c)	129.	(b, c, d)	130.	(a, b, c)	131.	(a, c)	132.	(a, b, d)				
133.	(b, c, d)	134.	(a, c)	135.	(b, c)	136.	(a, b, c)	137.	(b)								

Explanations Syntax Analysis (Parser)

5. (a)

The string *abbcde* can be obtained using RMD by following steps

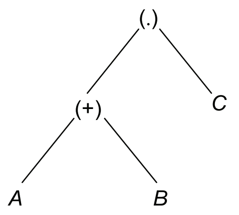
$S \to aABe \to aAde \to aAbcde \to abbcde.$

10. (d)

The grammar is ambiguous because the string '*b*' has two parse trees as shown below.

$$\begin{tikzcd}
S \arrow[d] & S \arrow[d] \\
A \arrow[d] & B \arrow[d] \\
b & b
\end{tikzcd}$$

18. (a)

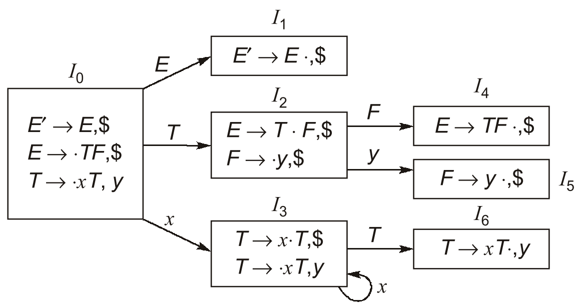


By applying inorder traversal we have $(A + B) * C$.

20. (b)

Handle pruning is the technique used in bottom to up parsing i.e. canonical reduction sequence.

22. (7)



Above LALR construction has 7 states.

24. (b)

In this, first we draw the goto graph with LR(0) item then find the number of inadequate state so we get