

1. Compiler translates the source code to
 - a. Executable code
 - b. Machine code
 - c. Binary code
 - d. **Both B and C**
2. Which of the following groups is /are token together into semantic structures?
 - a. Syntax analyzer
 - b. Intermediate code generation
 - c. **Lexical analyzer**
 - d. Semantic analyzer
3. Compiler should report the presence of _____ in the source program, in translation process.
 - a. Classes
 - b. Objects
 - c. **Errors**
 - d. Text
4. What is the output of lexical analyzer?
 - a. A parse tree
 - b. **A list of tokens**
 - c. Intermediate code
 - d. Machine code
5. Grammar of the programming is checked at _____ phase
 - a. Semantic analysis
 - b. **Syntax analysis**
 - c. Code optimization
 - d. Code generation
6. _____ is a process of finding a parse tree for a string of tokens.
 - a. **Parsing**
 - b. Analyzing
 - c. Recognizing
 - d. Tokenizing
7. What is the action of parsing the source program into proper syntactic classes?
 - a. **Lexical analysis**
 - b. Syntax analysis
 - c. General Syntax analysis
 - d. Interpretation analysis
8. Compiler can check _____ error.
 - a. Logical
 - b. **Syntax**
 - c. Content
 - d. A and B
9. A grammar that produces more than one parse tree for some sentence is called as
 - a. **Ambiguous**
 - b. regular
 - c. Unambiguous
 - d. All of these
10. Lexical analysis is about breaking a sequence of characters into
 - a. Groups
 - b. Packets
 - c. Lines
 - d. **Tokens**
11. _____ is considered as a sequence of characters in a token.

- a. Texeme
 - b. Pattern
 - c. Lexeme**
 - d. Mexeme
12. A _____ is a software utility that translates code written in higher language into a low level language.
- a. Converter
 - b. Compiler**
 - c. Text editor
 - d. Code optimizer
13. In a compiler, keywords of a language are recognized during
- a. The code generation
 - b. Parsing of the program
 - c. The lexical analysis of the program**
 - d. Data flow analysis
14. Which of the following component is important for semantic analysis?
- a. Yacc
 - b. Lex
 - c. Type checking**
 - d. Symbol table
15. Which phase of the compiler is also known as Scanner?
- a. Syntax analysis
 - b. Lexical analysis**
 - c. Semantic analysis
 - d. Code generation
16. Which phase of the compiler is also known as Parser?
- a. Syntax analysis**
 - b. Lexical analysis
 - c. Semantic analysis
 - d. Code generation
17. Keywords are recognized in a compiler during –
- a. The code generation
 - b. The data flow analysis
 - c. The lexical analysis of the program**
 - d. The program parsing
18. Which statement is an abstract form of intermediate code?
- a. Three address**
 - b. Two address
 - c. Address
 - d. Intermediate code
19. The function _____ is automatically generated by the flex when it is provided with a .l file
- a. flex()

b. yylex()

c. lex()

d. nnlex()

20. In the input file, In which section Anything written in this brackets is copied directly to the file lex.yy.c?

A. Rules Section

B. Definition Section

C. User Code Section

D. None of the above

21. The rule section is enclosed in?

A. %%

B. \$\$

C. %%%%

D. \$\$\$\$

22. State whether following grammar in operator precedence grammar or not?

$E \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

a. Yes

b. No

23. State whether following grammar in operator precedence grammar or not?

$E \rightarrow EOE$

$E \rightarrow id$

$O \rightarrow + | * | /$

a. Yes

b. No

24. State whether following grammar in operator precedence grammar or not?

$E \rightarrow E + E | E * E | E / E | id$

a. Yes

b. No

25. State whether following grammar in operator precedence grammar or not?

$S \rightarrow SAS | a$

$A \rightarrow bSb | b$

a. Yes

b. No

26. In LR(k) parsing L stands for

- a. **left to right scanning**
- b. right-most derivation
- c. right to left scanning
- d. lookahead

27. In LR(k) parsing L stands for

- a. left to right scanning
- b. **right-most derivation**
- c. right to left scanning
- d. lookahead

28. In LR(k) parsing L stands for

- a. left to right scanning
- b. right-most derivation
- c. right to left scanning
- d. **k lookahead**

29. If state does not know whether it will make a shift operation or reduction for a terminal, we say that there is a shift/reduce conflict.

- a. **True**
- b. False

30. state does not know whether it will make a reduction operation using the production rule I or j for a terminal, we say that there is a_____

- a. shift/reduce conflict
- b. **reduce/reduce conflict**
- c. shift/ shift conflict
- d. none of the above

31. LALR stands for...

- a. **Lookahead LR**
- b. Left Rule LR
- c. Lookahead Left Rule
- d. All of the above

32._____ creates a LALR paesers for the given grammar.

- a. **yacc**
- b. lex
- c. yacc and lex
- d. none of the above

33. The general idea is to use the FIRST & FOLLOW too construct the parsing table.

- a. **True.**

B False

34. Which parser is most powerful in the following parsers?

- a. Operator Precedence
- b. SLR
- c. Canonical LR**
- d. LALR

35. The output of the lexical analyzer is _____

- a. string character
- b. a syntax tree
- c. a set of RE
- d. a set of tokens**

36. Which grammar gives multiple parse trees for the same string?

- a. Unambiguous
- b. Regular
- c. Ambiguous**
- d. All of the above

37. Which of the following parser is a top-down parser?

- a. An LALR parser
- b. A LR parser
- c. Operator precedence parser
- d. Recursive descent parser**

38. Which parser is known as the shift-reduce parser?

- a. Bottom-up parser**
- b. Top-down parser
- c. Both Top-down and bottom-up

d. None of the Above

39. Which of the following function is called the canonical collection of LR(0) item.

a. FIRST

b. GOTO

c. COMPUTE

d. FOLLOW

40. Which of the following option is not a function of the shift-reduce parser?

a. Reduce

b. Accept

c. Go

d. Shift

41. Which of the following grammar has no two adjacent non-terminals?

a. Irregular grammar

b. Regular grammar

c. Operator

d. Precedence grammar

42. In which parsing, the parser constructs the parse tree from the start symbol and transforms it into the input symbol.

a. Bottom-up parsing

b. Top-down parsing

c. None of the above

d. Both a and b

43. Which derivation is generated by the top-down parser?

a. Right-most derivation in reverse

b. Left-most derivation in reverse

- c. Right-most derivation
- d. Left-most derivation**

44. Which derivation is generated by the bottom-up parser?

- a. Right-most derivation in reverse**
- b. Left-most derivation in reverse
- c. Right-most derivation
- d. Left-most derivation