1. In a two pass assembler the object code generation is done during the ?

a) Second pass

b) First pass

c) Zeroeth pass

d) Not done by assembler

Answer: a

2. Which of the following is not a type of assembler ?

a)

b)

c)

d)

Answer:

Explanation:

3. In a two pass assembler, adding literals to literal table and address resolution of local symbols are done using ?

a) First pass and second respectively

b) Both second pass

c) Second pass and first respectively

d) Both first pass

Answer: d

4. In a two pass assembler the pseudo code EQU is to be evaluated during ?

a) Pass 1

b) Pass 2

c) not evaluated by the assembler

d) None of above

Answer: a

5. Which of the following system program foregoes the production of object code to generate absolute machine code and load it into the physical main storage location from which it will be executed immediately upon completion of the assembly ?

a) Macro processor

b) Load and go assembler

c) Two pass assembler

d) Compiler

Answer:

Explanation:

6. The disadvantage of machine level programming is

a) time consuming

b) chances of error are more

c) debugging is difficult

d) all of the mentioned

Answer: d

Explanation: The machine level programming is complicated.

7. The coded object modules of the program to be assembled are present in

a) .ASM file

b) .OBJ file

c) .EXE file

d) .OBJECT file

Answer: b

Explanation: .OBJ file is created with same name as source file and extension .OBJ. It contains the coded object modules of the program to be assembled.

8. The advantages of assembly level programming are

a) flexibility of programming is more

b) chances of error are less

c) debugging is easy

d) all of the mentioned

Answer: d

Explanation: The assembly level programming is more advantageous than the machine level programming.

9. The extension that is essential for every assembly level program is

a) .ASP

b) .ALP

c) .ASM

d) .PGM

Answer: c

Explanation: All the files should have the extension, .ASM.

10. The directory that is under work must have the files that are related to

a) Norton’s editor

b) Assembler

c) Linker

d) All of the mentioned

Answer: d

Explanation: Before starting the process of entering a small program on PC, ensure that all the files namely Norton's editor, assembler, linker and debugger are available in the same directory in which work is been done.

11. The listing file is identified by

a) source file name

b) extension .LSF

c) source file name and an extension .LSF

d) source file name and an extension .LST

Answer: d

Explanation: The listing file is automatically generated in the assembly process and is identified by the entered or source file name and an extension .LST.

12. The extension file that is must for a file to be accepted by the LINK as a valid object file is

a) .OBJ file

b) .EXE file

c) .MASM file

d) DEBUG file

Answer: a

Explanation: The .OBJ extension is a must for a file to be accepted by the LINK as a valid object file.

13. The listing file contains

a) total offset map of a source file

b) offset address and labels

c) memory allotments for different labels

d) all of the mentioned

Answer: d

Explanation: The listing file contains total offset map of source file including labels, offset addresses, opcodes, memory allotments for different directives and labels and relocation information.

14. DEBUG.COM facilitates the

a) debugging

b) trouble shooting

c) debugging and trouble shooting

d) debugging and assembling

Answer: c

Explanation: DEBUG.COM is a DOS utility that facilitates the debugging and trouble shooting.

15. DEBUG is able to troubleshoot only

a) .EXE files

b) .OBJ files

c) .EXE file and .OBJ file

d) .EXE flie and .LST file

Answer: a

Explanation: The DEBUG may be used either to debug a source program or to observe the results of execution of an .EXE file.

16. In a compiler the module that checks every character of the source text is called _____

a) The code generator

b) The code optimizer

c) The lexical analyzer

d) The syntax analyzer

Answer: a

Explanation: Lexical analysis is the process of converting a sequence of characters into a sequence of tokens.

17. The context free grammar is ambiguous if _____

a) The grammar contains non-terminals

b) Produces more than one parse tree

c) Production has two non-terminals side by side

d) None of the mentioned

Answer: b

Explanation: Since more than one parse tree is generated hence one than option is available .Therefore it’s ambiguous.

18. What is another name for Lexical Analyser?

a) Linear Phase

b) Linear Analysis

c) Scanning

d) All of the mentioned

Answer: d

Explanation: Lexical Analyzer is also called “Linear Phase” or “Linear Analysis” or “Scanning“.

19. An individual token is called _____

a) Lexeme

b) Lex

c) Lexeme & Lex

d) None of the mentioned

Answer: a

Explanation: Individual Token is also Called Lexeme.

20. Lexical Analyser’s Output is given to Syntax Analysis.

a) True

b) False

Answer: a

Explanation: Lexical Analyzer’s Output is given to Syntax Analysis.

21. Which phase of the compiler is Lexical Analyser?

a) First

b) Second

c) Third

d) None of the mentioned

Answer: a

Explanation: Lexical Analyzer is First Phase of Compiler.

22. Input to Lexical Analyser is _____

a) Source Code

b) Object Code

c) Lexeme

d) None of the mentioned

Answer: a

Explanation: Lexical analyser’s Input is Source Code.

23. Lexical Analysis Identifies Different Lexical Units in a _____

a) Source Code

b) Object Code

c) Lexeme

d) None of the mentioned

Answer: a

Explanation: Lexical Analysis Identifies Different Lexical Units in a source Code.

24. Which one is a type of Lexeme?

a) Identifiers

b) Constants

c) Keywords

d) All of the mentioned

Answer: d

Explanation: All of them along with Operators are different types of lexeme

25. A _____ is a string of characters which form a syntactic unit.

a) Lexeme

b) Lex

c) Lexeme & Lex

Answer: a

Explanation: A lexeme is a string of characters that form a syntactic unit.

26. Which of the following derivations does a top-down parser use while parsing an input string?

a) Leftmost derivation

b) Leftmost derivation in reverse

c) Rightmost derivation

d) Rightmost derivation in reverse

Answer: a

Explanation: In top down parser takes input from Left to right constructing leftmost derivation of the sentence.

27. 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?

a) Assembly

b) Parsing

c) Relocation

d) Symbol resolute

Answer: c

Explanation: Relocation is the process of replacing symbolic references or names of libraries with actual usable addresses in memory before running a program. Linker performs it during compilation.

28. Which of the following statements is false?

a) Left as well as right most derivations can be in Unambiguous grammar

b) An LL (1) parser is a top-down parser

c) LALR is more powerful than SLR

d) Ambiguous grammar can’t be LR (k)

Answer: a

Explanation: If a grammar has more than one leftmost (or rightmost) derivation the grammar is ambiguous. Sometimes in unambiguous grammar the rightmost derivation and leftmost derivations may differ.

29. Which of the following grammar rules violate the requirements of an operator grammar?

(i) P -> QR

(ii) P -> QsR

(iii) P -> ε

(iV) P -> QtRr

a) (i) only

b) (i) and (iii) only

c) (ii) and (iii) only

d) (iii) and (iv) only

Answer: b

Explanation: An operator precedence parser is a bottom-up parser that interprets an operator-precedence grammar.

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

A -> A1 #B {A.value = A1.value * B.value}

| B {A.value = B.value}

B-> B1 & F {B.value = B1.value + C.value}

|C {B.value= C.value }

C -> num {C.value = num.value}.

30. 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

Answer: c

Explanation: Higher precedence operator will never produce an expression with operator with lower precedence.&># in terms of precedence order.

31. Consider a program P that consists of two source modules M1(contains reference to a function defined in M2) and M2 contained in two different files.

a) Edit time

b) Compile time

c) Link time

d) Load time

Answer: c

Explanation: Compiler transforms source code into the machine language which is in binary.

Kinds of object codes:

i. Defined symbols, which allow it to be called by other modules,

ii. Undefined symbols, which call the other modules where these symbols are defined, and

iii. Symbols which are used internally within object file for relocation.

32. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?

a) Removing left recursion only

b) Factoring the grammar alone

c) Factoring & left recursion removal

d) None of the mentioned

Answer: d

Explanation: Factoring as well as left recursion removal do not suffice to convert an arbitrary CFG to LL(1) grammar.

33. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G has n2 states.

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 mentioned

Answer: b

Explanation: SLR parser has less range of context free languages than LALR but still both n1 & n2 are same for SLR & LALR respectively.

34. Match the following.

P. Regular expression 1. Syntax analysis

Q. Pushdown automata 2. Code generation

R. Dataflow analysis 3. Lexical analysis

S. Register allocation 4. Code optimization

a) P-4. Q-1, R-2, S-3

b) P-3, Q-1, R-4, S-2

c) P-3, Q-4, R-1, S-2

d) P-2, Q-1, R-4, S-3

Answer: b

Explanation: Syntax analysis has Regular expressions. The code optimization goes hand in hand with data flow analysis. Whereas CFG is related to PDA which is related to syntax analysis Register allocation is used in reference with code generation.

35. Inherited attribute is a natural choice in _____

a) Variable declarations record is maintained

b) L values and R values

c) All of the mentioned

d) None of the mentioned

Answer: a

Explanation: It keeps track of variable.

36. YACC builds up _____

a) SLR parsing table

b) Canonical LR parsing table

c) LALR parsing table

d) None of the mentioned

Answer: c

Explanation: It is a parser generator.

37. In an absolute loading scheme which loader function is accomplished by assembler?

a) Re-allocation

b) Allocation

c) Linking

d) Loading

Answer: a

Explanation: Large number variables onto a small number of CPU register.

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

a) Detects errors

b) None of the mentioned

c) Errors are passed to the text phase

d) All of the mentioned

Answer: c

Explanation: Advantage for a valid prefix property.

39. 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

Explanation: Conversion of characters to tokens.

40. A top down parser generates _____

a) Rightmost Derivation

b) Right most derivation in reverse

c) Left most derivation

d) Left most derivation in reverse

Answer: c

Explanation: Top-down parsing is a parsing strategy where one first looks at the highest level of the parse tree and works down the parse tree by using the rewriting rules of a formal grammar.

41. The below 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?

E -> number Eval number val

E E .val E .VAL E .val

E # E E .val E .VAL E .val

;

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

Answer: c

Explanation: Yacc 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.

42. 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?

E -> number Eval number val

E E .val E .VAL E .val

E # E E .val E .VAL E .val

;

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

Answer: b

Explanation: 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.

43. Consider the following grammar.

S -> S * E

S -> E

E -> F + E

E -> F

F -> id

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

(i) S -> S * .E

(ii) E -> F. + E

(iii) E “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?

a) (ii)

b) (i) and (iii)

c) (iii)

d) None of the mentioned

Answer: C

Explanation: If S -> S): E is in LR (0) then E -> F +: E will also be there because both of them has ‘: ‘ before E.

44. Consider the following grammar:

S -> FR

R -> * S | ε

F -> id

In the predictive parser table, M, of the grammar the entries M [S, id] and M [R, $] respectively.

a) {S ” FR} and {R ” ε}

b) {S ” FR} and {}

c) {S ” FR} and {R ” * S}

d) {F ” id} and {R ” ε}

Answer: a

Explanation: The predictive parser table is given as. Non Terminal) id $ S S “FR F F “id R R “) S R “! R “! So at M [S, id] = { S ” FR} M [R,$] = {R “!}.

45. Consider the following translation scheme.

S -> ER

R -> * E{print{’ * ’);

R | f

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 * 3 + 4

b) 2 * + 3 4

c) 2 3 * 4 +

d) 2 3 4 + *

Answer: b

Explanation: Input string 2) 3 + 4 S ” ER FR idR {print(2)} id)ER {print())} id) F+ER {print(+)}id) id + ER {print(3)} id) id) id +id So 2)+ 3 4 are printed.

46. 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

Answer: d

Explanation: All the statements are true except option (There is scope of dead code elimination in this code) since there is no dead code to get eliminated.

47. Which one of the following grammars generates the language L = (a i b i | i ! j}?

a)

S ->AC | CB

b)

S -> aS | Sb | a | b

C -> aCb | a | b

A -> aA | ε

B -> Bb | ε

c)

S -> ACCB

d)

S -> AC | CB

C -> aCb |!

C -> aCb |!

A -> aA |!

A -> aA | a

B -> Bb |!

B -> bB | b

Answer: d

Explanation: The grammar S -> AC CB

C ->aCb !

A ->aA a

B->” bB b

Consider string aaabb S -> AC AaCb AaaCbb Aaabb aaabb But string aabb S ” AC And ->his string is not derivable

48. Relocating bits used by relocating loader are specified by _____

a) Relocating loader itself

b) Linker

c) Assembler

d) Macro Processor

Answer: b

Explanation: Takes an object files and combines them into a single executable file, library file, or another object file.

49. In the correct grammar above, what is the length of the derivation (number of steps starting from S to generate the string a l b m with l ! m?

a) max (l, m) + 2

b) l+m+2

c) l + m + 3

d) max (l, m) + 3

Answer: a

Explanation: It is very clear from the previous solution that the no. of steps required depend upon the no. of a' s & b ' s which ever is higher & exceeds by 2 due to S " AC CB & C "! So max(l , m) + 2.

50. 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

Answer: a

Explanation: Clearly LR & LALR are not top down they are bottom up passers. Also not operator precedence parser. ut yes recursive descent parser is top down parser. Starts from start symbol & derives the terminal string.