CS606-Compiler Construction
FINAL TERM MCQS
Prepared by: JUNAID

AL-JUNAID TECH INSTITUTE

1. _____convert the relocatable machine code into absolute machine code by linking library and relocatable objectfiles.

   - Assembler
   - **Loader/link-editor**
   - Compiler
   - Preprocessor

2. Parsers take__as input from lexical analyzer.
   - Linker
   - **Token**
   - Instruction
   - None of the given

3. The regular expression__denotes, the set of all strings of a's and b's of length two
   - a*
   - (a*|b*)*
   - (a*b*)*
   - **(a|b)(a|b)**

4. _____is a regular expression for the set of all strings over the alphabet {a} that has an even number of a's.
   - **aa***
   - (aa)*
   - aa*a
   - a(aa)*

5. _____Phase supports macro substitution and conditional compilation.
   - Semantic
   - Syntax
   - **Preprocessing**
   - None of given

6. In LL(1) parsing algorithm,__contains a sequence of grammar symbols.
   - **Stack**
   - Link List
   - Array
   - None of the given.

7. Consider the grammar

A → B C D
B → h B | €
C → C g | g | C h | i
D→ AB | £

First of A is_____.

- **h, g, i**
- g
- h
- None of the given.

8. _____parsers never shifts into an error state.
   - LS
   - LT
   - **LR**
   - LP
9. In parser, the two LL stand for___.
   - Left-to-right scan of input
   - left-most derivation
   - **Left-to-right scan of input and left-most derivation** **PG # 54**
   - None of the given
10. is elaborated to produce bindings.
    - **Declaration**
    - Expression
    - Command
    - None of the given
11. A lexical analyzer generated by___is essentially a FSA.
    - Dex
    - Mex
    - Fex
    - **Lex**
12. A lexical analyzer generated by **lex** is essentially a PDA (Push Down Automaton).
    - True
    - **False**
13. T he actions (shift, reduce) in a SLR(1) parser depend on a look ahead

symbol (_____)

- **Current input token**
- Next Input Token
- Previous output Token
- Previous Input Token.

14. The following grammar contains a conflict. S → *A* | *xb*

- **Shift-Reduce**
- First-Reduce
- Shift-First
- Reduce-Reduce

15. S → A | xb
A → aAb | x
This grammar contains a_____conflict.

- **Shift-Reduce**
- First-Reduce
- Shift-First
- Reduce-Reduce

16. Consider the Following
S → AB

- 1
- **2**
- **3**
- 4

17. ___is a register allocation technique that *always* finds the minimal number of registers needed for aprocedure.

- Dangling reference
- **Graph coloring**
- Left Factoring
- Right Recursion

18. **Graph coloring** is a register allocation technique that operates at *individual* basic blocks.

- True
- **False**

19. **Graph coloring** is a register allocation heuristic that *usually*

finds he minimal number of registers needed for aprocedure.

- **True**
- False

20. S → a S | Sa | c
This grammar is ambiguous.

- **True**
- False

21. When generating code at the basic block level, the dependency graph must be converted to target code. By identifying , instruction selection and instruction ordering can be performed efficiently in a single pass.

- **Ladder sequences**
- Physical sequences
- Logical sequences
- Token sequences

22. _can be considered a small compiler since it transforms a source language (assembly) into a lessabstract target language (binary object code)

- Parser
- Assembler
- **Lexical analyzer**
- Scanner

23. When memory allocator operates on chunks which include some administrative part and a block of user data. Theadministrative part includes___flag for marking the chunk as free or in-use.

- **One**
- Two
- Three
- Four

24. parser transforms a stream of tokens into an_____.

- **AST**
- IST

- EST
- ATS

25. The parser generator yacc can handle_____grammars
- **LL(1)**
- LT(1)
- LS(1)
- LF(1)

26. The parser generator **yacc** can handle LL(1) grammars.
- **True**
- False

27. The **yacc** parser generator can handle LALR(1) grammars.
- **True**
- False

28. Simple code generation considers one AST node at a time. If the target is a *register* machine, the code can be generated in one__traversal of the AST, possibly introducing temporaries when running out of registers.
- **Depth-first**
- Breadth-first
- Depth-second
- Breadth-second

29. A linker combines multiple object files into a_____executable object.
- **Single**
- Double
- Triple
- Quadruple

30. The notation____instructs YACC to push a computed attribute value on the stack.
- **$$**
- &&
- ##
- --

PG # 106

31. The following two items

A → P • Q

B → P • Q
can co-exist in an___item set

- **LR**
- LS
- LT
- PR

32. When generating a lexical analyzer from a_____description, the item sets (states) are constructed by twotypes of "moves": character moves and € moves.

- Character
- Grammar
- **Token**
- Sentence

33. Hybrid IRs combine elements of____

- Graphical (structural)
- Linear IRs
- **Both graphical and linear IRs** **PG # 108**
- Non-Linear IRs

34. x[i] = y This is___.

- Prefix assignment
- Postfix assignment
- **Index assignment** **PG # 115**
- Non-Index assignment

35. A lexical analyzer generator automatically constructs a_____that recognizes tokens.

- **FA** **PG # 18**
- PDA
- DP
- Unidirectional Graph

36. if x relop y goto L Above statement is _____

- Abstract jump
- **Conditional jump** **PG # 115**

- While loop
- Unconditional jump

37. I

n a CFG (Context Free Grammar) the set of terminal and non-terminal symbols must be.

- **Disjoint**
- Logical
- Relational
- Joint

38. S → a | B

B → Bb | €

The non-terminal B is left recursive.

- **True**
- False

39.

YACC contains built-in support for handling ambiguous grammars resulting in conflicts.

- **Shift-reduce**
- Shift-Shift
- Reduce-reduce
- Reduce-Shift
- Segment-directed

43.When constructing an LR(1) parser we record for each item exactly in which context it appears, which resolvesmany conflicts present in_ parsers based on FOLLOW sets.

- **SLR(1)**
- LRS(1)
- RLS(1)
- SLL(1)

44.Code generation module has to tackle____.

- Memory management
- Instruction selection
- Instruction scheduling
- **All of the given** **PG # 129**

45. For convenience, lexical analyzers should read the complete__program

into memory.

- **Input**
- Output
- Input and output
- Tokens

40. Considering the following grammar:

S → A | x
A → aAb | x
The grammar contains a___conflict.

- **Reduce-reduce**
- First-first
- Shift-shift
- Shift-reduce

41. SLR (1) parsers only reduce a production rule when the current input token is an element of the FOLLOW set ofthat rule.

S → A B

A → € | aA

A → b | bB

- FOLLOW (A) contains 2 elements.

- True
- **False**

42. SLR (1) parsers only reduce a production rule when the current input token is an element of the FOLLOW set ofthat rule.

S → A B
A → a | aA
B→€|bB

- FOLLOW (A) contains 2 elements.

- **True**
- False

43. The order in which the DAG is traversed can lead to_____code

- **Better** **PG # 143**
- Worse
- Large
- Garbage

44. Register allocation problem uses the strategy of_____.

- **Graph coloring** **PG # 144**
- Graph nodding
- Graph edging
- Graph patching

48. Typical compilation means programs written in high-level languages to low-level

- **Object code** **PG # 06**
- Byted code
- Unicode
- Object code and byte code

45. In compilation process, Hierarchical analysis is also called_.

- Parsing
- **Syntax analysis.**
- Parsing and syntax analysis
- None of the given

46. IR (Intermediate Representation) stores the value of its operand in___.

- **Registers** **PG # 10**
- Memory
- Hard disk
- None of the given

47. exeme is a sequence of characters in the source program that is matched by the pattern for a.

- Linker

- **Token**
- Control flow
- None of the given

48. Parsers take___ as input from lexical analyzer.

- Linker
- **Token**
- Instruction
- None of the given

49. What kind of abstract machine can recognize strings in a regular set?

- **DFA**
- NFA
- PDA
- None of the given

53. In DFA minimization, we construct one__for each group of states from the initial DFA.

- **State** **PG # 30**
- NFA
- PDA
- None of the given

50. _____(Lexical Analyzer generator), is written in java.

- Flex
- **Jlex** **PG # 31**
- Complex
- None of the given

51. In Flex specification file, different sections are separated by____.

- **%%** **PG # 31**
- &&
- ##
- None of the given

52. Recursive__parsing is done for LL(1) grammar.

- **Decent**
- Ascent

- Forward
- None of the given

56. Alternative of the backtrack in parser is Look ahead symbol in__.

- **Input**
- Output
- Input and output
- None of the given

53. Parser takes tokens from scanner and tries to generate_____

- Binary search tree
- **Parse tree**
- Binary search tree and parse tree.
- None of the given

54. In predictive parsing table, the rows represents_____.

- Terminals
- Both non-terminal and terminal
- **Non-terminal** **PG # 62**
- None of the given

55. A **predictive parser** is a **top-down** parser.

- **True**
- False

56. In LL(1) parsing algorithm, contains a sequence of grammar symbols.

- **Stack** **PG # 62**
- Link list
- Array
- None of the given

57. Bottom-up parsing uses only___kinds of actions.

- **Two** **PG # 71**
- Three
- Four
- Five

58. Bottom-up parsers handle a_class grammar.

- **Large** **PG # 49**
- Small
- Medium
- None of the given

59. The shift action a terminal on the stack.

- **Pushes** **PG # 73**
- Pops
- Both push and pops
- None of the given

60. Reduce action__zero or more symbols from the stack.

- Pushes
- **Pops** **PG # 73**
- Both push and pops
- None of the given

61. In compilers, linear analysis is also called_____.

- Lexical analysis
- Scanning
- **Lexical analysis and scanning**
- None of the given

62. Back End of two-pass compiler uses_____algorithm.

- O(n)
- O(n log n)
- **NP Complete**
- None of the given

63. The Back End of a compiler consist of_____.

- **Instruction selection**
- Register allocation

- Instruction scheduling
- All of the given

64. In Back End module of compiler, optimal register allocation uses__.
- O(log n)
- O(n log n)
- **NP-Complete**
- None of the given

65. lexeme is a sequence of characters in the source program that is matched by the pattern for a.
- Linker
- **Token**
- Control flow
- None of the given

66. _____is a regular expression for the set of all strings over the alphabets **{a}** that has an even number of **a**'s.
- **aa***
- (aa)*
- aa*a
- a(aa)*

67. ____algorithm is used in DFA minimization.
- James's
- Robert's
- **Hopcroft's** **PG # 25**
- None of the given

68. _is an important component of semantic analysis.
- Code checking
- **Type checking** **PG # 39**
- Flush checking
- None of the given

69. In , certain checks are performed to ensure that components of a program fit together meaningfully.
- Linear analysis
- Hierarchical analysis
- **Semantic analysis**

- None of the given

70. _____read the input character and produce sequence of tokens as output.
    - **Lexical analyzer**
    - Parser
    - Symbol table
    - None of the given
71. _____of a two-pass compiler is consist of instruction selection, Register allocation and instruction scheduling.
    - **Backend**
    - Frontend
    - Start
    - None of the given
72. _____is evaluated to yield a value.
    - Command
    - **Expression**
    - Declaration
    - None of the given
73. A parser transforms a stream of tokens into an AST (Abstract Syntax Tree).
    - **True**
    - false
74. A parser transforms a stream of characters into a stream of tokens.
    - True
    - **False**
75. A lexical analyzer transforms a stream of characters into a stream of tokens.
    - **True**
    - False
76. S → a | A
    A → Aa | a
    This grammar is ambiguous.
    - **True**
    - False

77. The regular expressions **(a+|b)?** and **a+|b?** describe the same set of strings.
    - **True**
    - False
78. The regular expressions **a*|b*** and **(a|b)*** describe the same set of strings.
    - True
    - **False**
79. The regular expressions **a+a** and **a*aa** describe the same set of strings.
    - **True**
    - False
80. A lexical analyzer *generator* automatically construct a FSA (Finite State Automaton) that recognizes tokens.The generator is driven by a **regular description**
    - **True**
    - False
81. The transition table in a lexical analyzer records for each state (row) which token, if any, is recognized in thatstate. - For each token there may be more than one "recognizing" row in the table.
    - **True**
    - False
82. A **recursive descent** parser is based on a PDA (Push Down Automaton).
    - **True**
    - False
83. A **bottom-up** parser creates the nodes in the AST in pre-order.
    - True
    - **False**
84. A **top-down** parser creates the nodes in the AST (Abstract Syntax Tree) in preorder.
    - **True**
    - False
85. A_____parser creates the nodes in the AST in preorder.
    - **Top – Down**
    - Bottom – Up

- Middle – Ware
- Straight

86. The stack used in a bottom-up parser contains an alternating sequence of states and grammar symbols.
    - **True**
    - False

87. The following two items
    A→ P• Q
    A→ PQ •
    Can coexist in an LR item set.
    - True
    - **False**

88. The Following two Items
A → x • B
B → • y
Can coexist in an LR item set.
- **True**
- False

89. The Following two Items
B → P • P
B → Q • Q
Can coexist in an LR item set.
- True
- **False**

90. S → A | xb
    A → aAb | x
    This is an LALR(1) grammar.
    - **True**
    - False

91. A **linker** combines multiple object files into a single executable object.
    - **True**
    - False

92. **Data-flow equations** can be solved efficiently by using bitwise boolean instructions (AND, OR, etc.).

- **True**
- False

93. **Data-flow equations** operate with IN, OUT, GEN, and KILL sets.
    - **True**
    - False
94. When **threading** an AST it might be necessary to introduce additional (join) nodes to ensure that each languageconstruct has a single exit point.
    - **True**
    - False
95. An **iterative** interpreter operates on a threaded AST.
    - **True**
    - False
96. S → A | B
    A → € | aA
    B → b | bB
    FIRST(S) contains_____elements.
    - 2
    - **3**
    - 4
    - None
97. The following set
    S → • A x {$}
    A → • a {x}
    A → • aA {x}
    is a valid LR(1) item set
    - True
    - False
98. S → Ab
    A → Aa | €
    - True
    - **False**
99. The regular expressions **a(b|c)** and **ab|ac** describe the same set of strings.
    - **True**

- False

100. S → a | B
B → Bb | E
The non-terminal___ is left recursive.

- **B**
- a
- E
- None of the given

101. In PASCAL_____ represent the inequality test.

- :=
- =
- **<>**
- None of the given

102. In parser the two LL stand(s) for_____.

- Left-to-right scan of input
- left-most derivation
- **All of the given**
- None of the given

103. Consider the grammar
A →B C D
B → h B | epsilon
C → C g | g | C h | i
D → AB | epsilon
First of C is_____.

- **g, I**
- g
- h , i
- i

104. Three-address codes are often implemented as a ___.

- **Set of quadruples** **PG # 104**
- Set of doubles
- Set of Singles
- None of the given

105. What does following statement represent? x[i] = y

- Prefix assignment
- Postfix assignment
- **indexed assignment** **PG #107**
- None of the given

106. _____convert the reloadable machine code into absolute machine code by linking library and reloadable objectfiles.

- Assembler
- **Loader/link-editor**
- Compiler
- Preprocessor

107. Consider the following grammar,

A → B C D
B → h B | epsilon
C → C g | g | C h | i
D → AB | epsilon

First of A is_____.

- **h, g, i**
- g
- h
- None of the given

108. One of the core tasks of compiler is to generate fast and compact executable code.

- **True** **PG # 14**
- False

109. Compilers are sometimes classified as.

- Single pass
- Multi pass
- Load and go
- **All of the given**

110. In multi pass compiler during the first pass it gathers information about

- **Declaration**

- Bindings
- Static information
- None of the given

111. We can get an LL(1) grammar by_____.

- Removing left recurrence
- Applying left factoring
- **Removing left recurrence and Applying left factoring**
- None of the given

112. Consider the following grammar, S --> aTUe T --> Tbc/b U --> d And suppose that string "abbcde" can be parsedbottom-up by the following reduction steps:

(i) aTbcde
(ii) aTde
(iii) aTUe
(iv) S

So, what can be a handle from the following?

- **The whole string, (aTUe)** **PG # 68**
- The whole string, (aTbcde)
- The whole string, (aTde)
- None of the given

113. When generating a lexical analyzer from a token description, the item sets (states) are constructed by two typesof "moves": character moves and_____moves.

- **E (empty string)** **PG # 18**
- #
- @
- none of given

114. Which of the following statement is true about Two pass compiler.

- Front End depends upon Back End

- **Back End depends upon Frond End** **PG # 5**
- Both are independent of each other
- None of the given

115. avoid hardware stalls and interlocks.
- Register allocation
- **Instruction scheduling** **PG #10**
- Instruction selection
- None of given

116. Front end of two pass compiler takes_____as input.
- **Source code** **PG # 5**
- Intermediate Representation (IR)
- Machine Code
- None of the Given

117. In Three-pass compiler _____is used for code improvement or optimization.
- Front End
- **Middle End** **PG # 10**
- Back End
- Both Front end and Back end

118. _____of a two-pass compiler is consists of Instruction selection, Register allocation andInstruction scheduling.
- **Back end** **PG # 9**
- Front end
- Start
- None of given

119. NFA is easy to implement as compared to DFA.
- True
- **False** **PG # 19**

120. In a transition table cells of the table contain the___state.

- Reject state
- **Next state** **PG #18**
- Previous state
- None of the given

121. The regular expressions a*|b* and (a|b)* describe the____set of strings.
- Same
- **Different**
- Onto

122. A canonical collection of sets of items for an augmented grammar, C is constructed as

- For each set / in C and each grammar symbol X where goto (C, X) is empty and not in C add the set goto (C, X)to C.
- **The first set in C is the closure of {[S' --> .S]}, where S' is starting symbol of original grammar and S is the starting non-terminal of augmented grammar.** **PG # 72**
- The first set in C is the closure of {[S' --> .S]}, where S is starting symbol of original grammar and S' is theStarting non-terminal of original grammar.

123. The_____translation statements can be conveniently specified in YACC
- **Syntax-directed** **PG # 120**
- Image-directed
- Sign-directed
- None of the given.

124. Attributes whose values are defined in terms of a node's own attributes, node's siblings and node's parent arecalled_.
- **Inherited attributes** **PG # 92**
- Physical attributes
- Logical attributes

- Un-synthesized attributes

125. Consider the grammar

A → B C D
B → h B | epsilon
C → C g | g | C h | i
D → AB | epsilon

Follow of B is_____.

- **h**
- g, h, i, $
- g, i
- g

126. Consider the grammar A --> B C D

A → B C D
B → h B | epsilon
C → C g | g | C h | i
D → AB | epsilon

Follow of C is_____.

- **g, h, i, $** **PG # 47**
- g, h, $
- h, i, $
- h, g, $

127. The test of string is described by a rule called a, associated with token.

- Character
- Loader
- **Pattern**
- None of the given

128. Bottom up parsing is also called____.

- **LR Parsing** **PG # 70**

- LT Parsing
- LS Parsing
- None of the given

129. A DFA can be reconstructed from NFA using the subset construction, similar to one used for

- **Lexical Analysis** **PG # 82**
- Physical Analysis
- Logical Analysis
- Parsing

130. Which of the following system software resides in the main memory always?

- Text editor
- Assembler
- Linker
- **Loader**

131. plays an important role in code optimization.

- **DAG** **PG # 143**
- Lexical Analyzer
- AGD
- Memory Management

132. LR parsers can handle_____grammars.

- **Left-recursive** **PG # 63**
- file-recursive
- End-recursive
- Start-recursive

133. Performing common sub expression elimination on a dependency graph requires the identification of nodeswith the same operator and operands. When using a hash table (with a hash

function based on operator and operands) all___nodes can be identified in linear time.

- **Common**
- Uncommon
- Next
- Previous

134. Linear IRs resembles pseudo-code for same__.

- Automated Machine
- Mechanical machines
- Token machines
- **Abstract machine** **PG # 100**

135. Responsibility of _____ is to produce fast and compact code.

- **Instruction selection**
- Register allocation
- Instruction scheduling
- **None of given Page no: 9**

136. Optimal registers allocation is an NP-hard problem.

- True
- **False** **Page no : 10**

137. Left factoring of a grammar is done to save the parser from back tracking.

- **True** **Page no:61**
- False

138. Recursive _____ parsing is done for LL(1) grammar.

- **Decent Page no : 47**
- Ascent
- Forward
- Backward

139. If X is a terminal in A--> aX•?, then this transition corresponds to a shift of ____ from input to top of parse stack.

- X
- A
- a
- None of the given

140. An ----- does not need to examin the entire stack for a handle, the state symbol on the top of the stack contains all the information it needs.

- LR parser
- RL parser
- BU parser
- None of the given

141. Suppose ? begins with symbol X which may be a terminal (token) or non-terminal. The item can be written as A? Xa•?.

- True
- False

142. YACC parser generator builds up

- SLR parsing table
- Canonical LR parsing table
- LALR parsing table
- None of the given

143. LR(1) parsing is --- base parsing.

- DFA
- CFG
- PDA
- None of the given

144. The LR(1) parsers can not recognize precisely those languages in which one-symbol lookahead suffices to determine whether to shift or reduce.

- True
- False

145. S --> A | xb A --> aAb | x This grammar contains a reduce-reduce conflict.

- **True**
- False

146. Following statement represents: if x relop y goto L

- **abstract jump**
- **Conditional Jump**

- **While Loop**
- **None Of Given**

147. Left factoring is enough to make a grammar LL(1).

- **True**
- **False**

148. S --> A B A --> e | aA B --> e | bB - FIRST(S) contains ___ elements.

- 3
- 4
- 5
- 6

149. Grammars with LL(1) conflicts can be made LL(1) by applying left-factoring, substitution, and left-recursion removal. Left-factoring takes care of _____conflicts.

- **FIRST/FIRST**
- **First/SECOND**
- **SECOND/FIRST**
- **NONE OF THE GIVEN**

150. In an attribute grammar each production rule(N--> a) has a corresponding attribute evaluation rule that describes how to compute the values of the _____attributes of each particular node N in the AST.

- Synthesized
- Complete
- Free
- Bound

151. When constructing an LR(1) parser we record for each item exactly in which context it appears, which resolves many conflicts present in _____parsers based on FOLLOW sets.

- SLR(1)
- LRS(1_
- RLS(1)

- None of the Given

152. Backpatching to translate flow-of-control statements in ____ pass.

- one
- two
- three
- all of the given

153. LR parsing _____ a string to the start symbol by inverting productions.

- **Reduce**
- **Shift**
- **Adds**
- **None of the Given**

154. _____ phase which supports macro substitution and conditional compilation.

- Semantic
- Syntax
- Preprocessing
- None of the Given

155. Parser always gives a tree like structure as output

- True
- False

156. **Lexer and scanner are two different phases of compiler**

- **True**
- **False**

157. _____tree in which each node represents an operator and children of the node represent the operands.

- **Abstract Syntax**
- **Concrete Syntax**
- **Parse**
- **None of the Given**

158. Register allocation by graph coloring uses a register interference graph. _____ nodes in the graph are joined by an edge when the live ranges of the values they represent overlap.

- **Two**
- **Three**
- **Four**
- **Five**

159. In compilation process Hierarchical analysis is also called
- Parsing
- Syntax Analysis
- Both Parsing and Syntax analysis
- None Of the Given

160. Ambiguity can easily be handled by Top-down Parser
- True
- False

161. Front-end of a two pass compiler is consists of Scanner.
- True
- False

162. LL(1) parsing is called non-predictive parsing.
- True
- False

163. In predictive parsing table the rows are _____.
- Non-Terminal
- Terminals
- Both A and B
- None of the Given

164. In LL1() parsing algorithm _____ contains a sequence of grammar symbols.
- Stack
- Link List
- Array

- None of the Given

165. AST summarizes the grammatical structure with the details of derivations.
- True
- False

166. f X is a non-terminal in A? aX•?, then the interpretation of this transition is more complex because non-terminals do not appear in input
- Yes
- No

167. If / is a set of items for a grammar then closure (/) is a set of items constructed from / by the following rule.

- If A --> aX.Y is in closure (/) and Y --> r is production, then add X --> .r to closure (/).
- If A --> a.XY is in closure (/) and X --> r is production, then add X --> .r to closure (/).
- If A --> aXY. is in closure (/) and A --> r is production, then add X --> .r to closure (/).
- None of these

168. NFA of LR(0) items means _____
- look-ahead one sybole
- no look-ahead
- look-ahead all sybols
- None of the given

169. A grammar is LR if a ------ shift reduce-reduce parser can recognize handles when they appear on the top of stack.
- left-to-reverse
- left-to-rise
- left-to-right
- None of the given.

170. The output from the algorithm of constructing the collection of canonical sets of LR(1) items will be the _____
- Original Grammar G
- Augmented grammar G'
- Parsing table
- None of the given

171. Reduction of a handle to the ------- on the left hand side of the grammar rule is a step along the reverse of a right most derivation.

- Terminal
- Non-terminal

172. NFA of LR(1) items means _____

- no look-ahead
- look-ahead one sybole
- look-ahead all sybols
- None of the given

173. performing common subexpression elimination on aa dependncy graph requires the identification of nodes with the same operator and operands.when using a hash table (with a hash function based on operator and operands) all_____ nodes can be identified in linear time.

- common
- uncommon
- next
- previous

174. Linear IRs resemble pseudo-code for same _____.

- Automated Machine
- Mechanical machines
- Token machines
- Abstract machine

175. The regular expressions a*|b* and (a|b)* describe the _____set of strings.

- Same
- **Different**
- Onto

176. Back patching to translate flow-of-control statements in ____ pass.

- **one Page no : 111**
- two
- three
- all of the given

177. Consider the following grammar, S --> aTUe T --> Tbc/b U --> d And suppose that string “abbcde” can be parsed bottom-up by the following reduction steps: (i) aTbcde (ii) aTde (iii) aTUe (iv) S So

what can be a handle from the following?

- The second (b) in (abbcde)
- The first (b) in (abbcde)
- The substring (cd) in (abbcde)
- None of the given

178. Yacc contains built-in support for handling ambiguous grammars resulting in _____ conflicts.

- **Shift-reduce**
- Shift-Shift
- Shift-second
- None of the given

179. The following two items A -> P • Q B -> P • Q can co-exist in an _____ item set.

- **LR**
- LS
- LT
- PR

180. The error handling mechanism of the yacc parser generator pushes the input stream back when inserting 'missing' tokens.

- **True**
- False

181. Flow of values used to calculate synthesized attributes in the parse tree is:

- **Bottom-up**
- Right to left
- Top-Down
- Left to right

182. A lexical analyzer transforms a stream of tokens. The tokens are stored into symbol table for further processing by the parser.

- **True**
- False

183. LR parsers can handle _____ grammars.

- **Left-recursive Page no: 163**
- file-recursive
- End-recursive
- Start-recursive

184. For each language to make LL(1) grammar, we take two steps,

1st is removing left recurrence and 2nd is applying fin sequence.

- True
- **False**

185. Can a DFA simulate NFA?

- Yes
- No
- **Sometimes**
- Depend upon nfa

186. Which of the statement is true about Regular Languages?

- Regular Languages are the most popular for specifying tokens.
- Regular Languages are based on simple and useful theory.
- Regular Languages are easy to understand.
- **All of the given**

187. The transition graph for an NFA that recognizes the language (a | b)*abb will have following set of states.

- {0}
- {0,1}
- {0,1,2}
- **{0,1,2,3}** **not sure**

188. Functions of Lexical analyzer are?

- Removing white space
- Removing constants, identifiers and keywords
- Removing comments
- **All of the given**

189. Consider the following grammar, S --> aTUe T --> Tbc/b U --> d And suppose that string “abbcde” can be parsed bottom-up by the following reduction steps: (i) aTbcde (ii) aTde (iii) aTUe (iv) S So, what can be a handle from the following?

- The whole string, (aTUe) Page no : 68
- The whole string, (aTbcde)
- The whole string, (aTde)
- None of the given

190. The LR(1) items are used as the states of a finite automaton (FA) that maintains information about the parsing stack and progress of a shift-reduce parser.

- **True Page no: 74**
- **False**

191. Flex is an automated tool that is used to get the minimized DFA (scanner).

- True
- False

192. We use ----- to mark the bottom of the stack and also the right end of the input when considering the Stack implementation of Shift-Reduce Parsing.

- Epsilon
- #
- **$**
- None of the given

193. When generating a lexical analyzer from a token description, the item sets (states) are constructed by two types of "moves": character moves and _____ moves.

- **E (empty string) Page no : 18**
- #
- @
- none of given

194. Let a grammar G = (Vn, Vt, P, S) is modified by adding a unit production S'--> S to the grammar and now starting non-terminals becomes S' and grammar becomes G' = (Vn U {S'}, Vt, PU{S' --> S}, S'). The Grammar G' is called the ----------

- **Augmented Grammar Page no : 76**
- Lesser Grammar
- Anonymous Grammar
- none of given

195. Parser takes tokens from scanner and tries to generate _____ .

- Binary Search Tree
- Parse Tree
- Syntax Trace
- None of the Given

196. In Flex specification file different sections are separated by ___.

- **%% Page no: 26**
- **&&**
- ##

- \\

197. In DFA minimization we construct one _____ for each group of states from the initial DFA.

- State Page no : 25
- NFA
- PDA
- None of Given

198. Intermediate Representation (IR) stores the value of its operand in _____ .

- Registers
- Memory
- Hard Disk
- Secondary Storage

199. In _____certain checks are performed to ensure that components of a program fit together meaningfully.

- Linear analysis
- Hierarchical analysis
- **Semantic analysis Page no : 33**
- None

199. A _____ is a top down parser.

- **Predictive Parsing**
- Reactive parser
- Proactive parser
- None of the given

200. Lexical Analyzer generator _____ is written in Java.

- Flex
- **Jlex Page no : 26**
- Complex
- None of given

201. _____avoid hardware stalls and interlocks.

- Register allocation
- **Instruction scheduling**
- Instruction selection
- None of given

202. Recursive _____ parsing is done for LL(1) grammar.

- **Decent**
- Ascent
- Forward
- Backward