1. A sentential From is ….
   1. Any regular expression modified by the Kleene start operator.
   2. Any terminal in a context-free grammar.
   3. The start symbol or any form derived from it in a context-free grammar.
   4. Any string generated by a regular expression.
2. Languages tend to differ greatly in \_\_\_\_ but often have very similar aspects of \_\_\_\_
   1. Computational structure ///// recursive hierarchy
   2. Syntax ///// semantics
   3. Compilation ///// interpretation
   4. Integers ///// floating point
3. A grammar specification includes all of the following\_except
   1. A set of terminal symbols
   2. A start symbol
   3. A stop symbol
   4. A set of production rules.
4. The Chomsky Language Hierarchy can be used to differentiate grammars on the basis of
   1. The storage required to recognize the grammar’s languages
   2. The level of automation required to recognize the grammar’s language
   3. The complexity of the language that can be expressed by the grammar.
   4. A and b
   5. A, b and c
5. To avoid ambiguity in a derivation from a context-free grammar
   1. The derivation must always be performed in a left-most fashion
   2. The derivation must always be performed in a right- most fashion
   3. The CFG must be defined so that in any derivation, only one parse tree can result
   4. A and c
   5. B and c
6. Regular expression are used to specify
   1. Token formats
   2. Context-free grammars.
   3. The Chomsky hierarchy.
   4. Expression syntax
7. Operator associatively is a \_\_\_\_\_\_ property of a programming language’s definition
   1. Syntactic
   2. Semantic
   3. Symbolic
   4. Lexical
8. Well-formed and/or nested parentheses, brackets etc are best recognized with
   1. A symbol table.
   2. A context-free grammar.
   3. A regular expression
   4. An association list
9. Operator precedence is usually expressed in a programming language’s …
   1. Context-free grammar
   2. Concurrency
   3. Regular expression
   4. Non-determinacy
10. Among other things, a regular expression can be
    1. The concatenation of two regular expressions
    2. A character.
    3. A regular expression modified by the Chomsky spleen operator
    4. A and b
    5. A, b and c
11. Context-free grammars are more expressive than regular expression.
    1. No regular expression are more expressive than context-free grammars.
    2. Yes
    3. No they have the same level of expressiveness
    4. Maybe. It depends on the order of the semantic hierarchy.
12. Grammars are categorized into classes based on
    1. The length of the strings that they can generate.
    2. The number of non-terminal symbols they are allowed to use
    3. Whether they can be compiled or interpreted.
    4. The forms of production rules they are allowed to used.
13. Operator precedence is …
    1. Expressed using the non-edulidean transform
    2. Organized into levels
    3. Represented by non-deterministic finite automata
    4. Defined via the Chomsky spleen theorem
14. A context free grammar is used to specify a language’s …
    1. Heap and stack allocation rules.
    2. Parse structure.
    3. Dynamic semantic checks
    4. Regular expression.
15. Semantics is the \_\_\_\_ as opposed to its\_\_\_\_\_
    1. Flexibility ///// efficiency
    2. Efficiency ///// flexibility
    3. Form ///// meaning
    4. Meaning ///// form
16. The null (or empty) string is a valid regular expression.
    1. False
    2. True
17. The Kleene plus operator extends regular expressions to the expressiveness of context-free grammars
    1. True
    2. False
18. A regular expression is not allowed to
    1. Concatenate, even alternately
    2. Alternate, even iteratively
    3. Recurse, even indirectly
    4. Repeat, even redundantly
19. The exact steps to execute a program must be \_\_\_\_ in a \_\_\_\_\_
    1. Regular ///// Carteslan 3D grid
    2. Determinable ///// predictable and repeatable fashion
    3. Non-deterministic ///// Backus-Naur name space
    4. Context free ///// Chomsky-derived hierarchy
20. Right-most and left-most derivations from a context-free grammar…
    1. Always produce the same parse tree
    2. Never produce the same parse tree.
    3. May produce the same parse tree. It depends on the grammar.
21. Regular expression and context-free grammars both occur in the Chomsky language Hierarchy. Therefore, ….
    1. Nothing special. They are just different level of expressivity.
    2. They take the same amount of time and effort to process
    3. They may be transformed into one-to-one equivalences.
    4. They both get to use the Chomsky spleen operator in their production rules.
22. In a regular expression, the ? operator represents…
    1. Zero or one occurrences of a regular expression.
    2. The negation of a regular language set.
    3. One or more repetitions of a regular expression
    4. Zero or more repetitions of a regular expression
23. In a regular expression, the Kleene Plus operator represents …
    1. The set of all strings generated by the complete regular expression
    2. One ore more repetitions of a regular expression
    3. The extension of a regular language set to that of a context-free grammar.
    4. The compactification of a regular language set.
24. Operator precedence is a \_\_\_\_\_ property of a programming language’s definition
    1. Lexical
    2. Semantic
    3. Symbolic
    4. Syntactic
25. A grammar specification includes all of the following\_except\_
    1. A set of production rules.
    2. A set of non-terminal symbols
    3. A set of regular expressions
    4. A set of terminal symbols
26. Because a given regular expression may be converted to a finite automation, its language
    1. Is always infinite.
    2. Cannot ever be infinite
    3. Must be finite
    4. Might be infinite depending on the regular expression
27. Numbers, identifiers, keywords, operators, and punctuation marks are different kinds of
    1. Tokends
    2. Symbol tables
    3. Parse trees
    4. Dynamic categories
28. Syntax is the \_\_\_\_ as opposed to its \_\_\_\_
    1. Form ///// meaning
    2. Efficiency ///// flexibility
    3. Meaning ///// form
    4. Flexibility ///// efficiency
29. Regular expressions cannot express strings that have
    1. Three or more alternating characters.
    2. Infinite occurrences of a single character.
    3. Arbitrary matching pairs
    4. Two alternating characters.
30. A “mechanical” computation is one that…
    1. Does not require human intelligence, intervention, or interpretation to proceed.
    2. Operates by interacting with a human operator as the computation progresses
    3. Requires the highest performing computing devices to complete
    4. Originally operated on primitive electro-mechanical hardware but has been updated.
31. Well-formed and / or nested parentheses, brackets, etc, are best recognized with
    1. A regular expression
    2. A context-free grammar.
    3. A symbol table.
    4. An association list
32. Right-most and left-most derivation always result in the same parse tree.
    1. Maybe, it depends on context-free grammar being used.
    2. No
    3. Yes
33. The expressiveness of regular expressions may be expended to that context-free grammars by adding
    1. The alternation of two regular expression
    2. The Chomsky spleen operator
    3. Recursion
    4. B and c
    5. A b and c
34. A sentence is a sentential from which has only
    1. Non-terminal symbols
    2. Regular expression structure
    3. Terminal symbols
    4. Alternation and concatenation but not any empty strings
35. Tokens are …
    1. The internal nodes of every program’s parse tree
    2. The shortest strings of characters in a source program that have individual mening
    3. The basic building blocks of programs
    4. B and c
    5. A, b and c
36. Regular expressions are used to specify
    1. Token formats
    2. Context-free grammars
    3. Expression syntax.
    4. The Chomsky hierarchy
37. Operator associativity is usually expressed in a programming language’s
    1. Context-free grammar
    2. Non-determinacy
    3. Regular expressions
    4. Concurrency.
38. The definition of regular expression is often extended by string-processing programming languages
    1. No
    2. Yes
    3. Maybe, it depends on the Chomsky language hierarchy
39. A context-free grammar is used to specify a language’s
    1. Parse structure
    2. Regular expression
    3. Dynamic semantic checks
    4. Heap and stack allocation rules.
40. Unlike natural languages, programming languages must be \_\_\_\_\_ so that \_\_\_\_\_
    1. Defined without, formal notation ///// the compiler may be constructed completely automatically
    2. As simple as possible ///// anyone can use them with no training
    3. As difficult as possible ///// programmers are seen as geniuses
    4. Unambiguous and precise ///// the computer knows what to do and the user knows what to expect.
41. A formal notation is one that is
    1. Sufficiently precise and well defined so that it can be processed mechanically
    2. Expressed using only the Kleene star operator
    3. Usable for regular expressions but not context free grammars
    4. Representable in binary form but not in ternary form
42. The set of strings defined by a regular expression is known as its
    1. Iteration
    2. Alternation
    3. Language
    4. Kleene star
43. A context-free grammar that permits multiple parse trees for the same string is …
    1. Right-most
    2. In normal sentential form
    3. Left-most
    4. Ambiguous
44. Lexical analysis converts \_\_\_\_ into \_\_\_\_\_
    1. Tokens ///// a parse tree
    2. A parse tree ///// an abstract syntax tree
    3. An abstract tree //// the target language
    4. Characters ///// tokens
45. Backus-Naur Form (BNF) is..
    1. A mathematical notations for representing floating-point numbers
    2. A logic calculus for expressing dynamic syntactic checks
    3. An informal notation for writing regular expressions
    4. A formal notation for writing context-free grammars
46. Grammars are categorized into classes based on
    1. The length of the strings that they can generate
    2. The forms of production rules they are allowed to use
    3. Whether they can be compiled or interpreted
    4. The number of non-terminal symbols they are allowed to use
47. A sentential form is
    1. The start symbol or any form derived from it in a context-free grammar
    2. Any regular expression modified by the Kleene star operator
    3. Any terminal ion a context-free grammar
    4. Any string generated by a regular expression
48. Operator associativity is …
    1. Either left-to-right, right-to-left. Or does not apply.
    2. Controlled by the application of the appropriate regular expression
    3. Distinguished on the basis of Backus-Naur levels
    4. Differentiate on the basis of the Chomsky language hierarchy
49. Formal notation is used to describe a language’s \_\_\_\_ and \_\_\_\_\_
    1. Rationale ///// restructuring
    2. Exceptions //// extensions
    3. Syntax //// semantics
    4. Purpose ///// paradigms
50. A sentence is a sentential form which has only
    1. Alternation and concatenation but not any empty string
    2. Non-terminal symbols
    3. Terminal symbols
    4. Regular expressions structure.
51. Syntactical analysis converts \_\_\_\_into\_\_\_\_
    1. Tokens /// a parse tree
    2. A parse tree //// an abstract syntax tree
    3. Characters //// tokens
    4. An abstract syntax tree /// the target language