

Q1. Lexical analyser generates the next token

- (A) When it is asked for
- (B) By maintaining a repository of tokens
- (C) By scanning the whole input
- (D) None of the other options

Ans: A

Q2. Lexical analysis and parsing are put as two different phases so as to

- (A) Make design simple
- (B) Improve efficiency
- (C) Enhance portability
- (D) All of the other options

Ans: D

Q3. A lexeme is

- (A) Any sequence of characters
- (B) Sequence of characters defining a token
- (C) Same as a token
- (D) Not related to any token

Ans: B

Q4. Character sequence "2r" is not a lexical error if the language allows

- (A) Symbol names to start with integers
- (B) Juxtaposition is taken as a valid operator
- (C) Length of symbol names is not fixed
- (D) Symbol names to start with integers or juxtaposition is taken as a valid operator

Ans: D

Q5. A certain compiler corrects errors like "fi" to "if" automatically. This is an example of recovery in

- (A) Panic mode
- (B) Delete character
- (C) Replace character
- (D) Transpose characters

Ans: D

Q6. For the Fortran language statement "DO 5 I = 1.25" returns token IDENTIFIER for DO 5 I after looking upto

- (A) I
- (B) =
- (C) .
- (D) 5

Ans: C

Q7. A regular expression represents

- (A) Constituent strings of a language
- (B) Part of a language
- (C) Cannot represent any language
- (D) None of the other options

Ans: A

Q8. The regular expression  $(0|1)^*(0|1)$  represents a language with

- (A) Nonempty binary strings
- (B) Empty and nonempty binary strings
- (C) Odd nonempty strings
- (D) Even nonempty strings

Ans: A

Q9. The regular expression  $(0|1)^*00$  will accept all strings

- (A) Divisible by 2
- (B) Divisible by 4
- (C) Divisible by 2 with minimum length 2
- (D) Divisible by 4 with minimum length 2

Ans: D

Q10. A regular expression for accepting strings with exactly one 1 more than 0's is

- (A)  $0^*1$
- (B)  $(0|1)^*1(0|1)^*$
- (C)  $(0|1)^*1(0|1)^*|1(0|1)^*$
- (D) Not possible

Ans: D

Q11. Finite automata is an implementation of

- (A) Regular expression
- (B) Any grammar
- (C) Part of a regular expression
- (D) None of the other options

Ans: A

Q12. A DFA cannot have

- (A) Epsilon transitions
- (B) An isolated state
- (C) More than one transition from a state on same input
- (D) All of the other options

Ans: D

Q13. Number of possible epsilon transitions from a state in an NFA is

- (A) One
- (B) At most one
- (C) Many
- (D) Zero

Ans: C

Q14. Between NFA and DFA which one is easier to implement

- (A) NFA
- (B) DFA
- (C) Equal effort needed
- (D) Cannot be said definitely

Ans: B

Q15. Between NFA and DFA which one is more powerful

- (A) NFA
- (B) DFA

(C) Equally powerful

(D) Cannot be said definitely

Ans: C