Name: Enrolment No:	UPES UNIVERSITY WITH A PURPOSE

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
End Semester Examination, July 2020

Course: Compiler Design **Semester: VI**
Program: B.Tech CS with SPZ in MT **Time 02 hrs.**
Course Code: CSEG-3015 **Max. Marks: 100**

Instructions: Attempt all Questions

1. In a compiler, keywords of a language are recognized during:
   a. Parsing of the program
   b. Code Generation
   c. Lexical Analyzer
   d. Code Optimization

2. How many derivation trees are there for string aaa given grammar G? $S \rightarrow aS \mid Sa \mid a$
   a. 3
   b. 4
   c. 5
   d. 6

3. An identifier is permitted to be a letter followed by any number of letter and digits. Which of the following expression defines an identifier:
   a. $(L+D)^*$
   b. $(L+D)^+$
   c. $L(L+D)^+$
   d. $L(L+D)^*$

4. The number of tokens in the following C statements are: printf("i=%d, &i=%x", i, &i);
   a. 10
   b. 3
   c. 21
   d. 26

5. Consider a program P having two source modules M1 and M2. If M1 contains a reference to a function defined in M2 then the reference will be resolved at
   a. Compile time
   b. Link time
   c. Run time
   d. Load time

6. Which of the following data structure is used for managing information about variables and their attributes:
   a. Parse table
   b. Code table
   c. Lexical table
   d. Symbol table

7. Match all items in Group 1 with correct options from those given in Group 2.

Group 1	Group 2
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-4, R-1, S-2
c. P-3, Q-1, R-4, S-2
d. P-2, Q-1, R-4, S-3

8. Which one of the following statements is FALSE?
   a. Type checking is done before parsing.
   b. High-level language programs can be translated to different intermediate representations.
   c. Context free grammar can be used to specify both lexical and syntax rules.
   d. Arguments to a function can be passed using the program stack.

9. Which of the following grammar is free from left recursion:
   a. S → AB, A → Aa | b, B →c
   b. S → Aa|B, A → Bd|Sc, B →d
   c. S → Aa|Bb, A → Bd, B →Ae
   d. S → AB|Bb|c, A → Bd, B →e

10. A compiler for a high-level language that runs on one machine and produces code for a different machine is called
    a. Optimizing compiler
    b. One pass compiler
    c. Cross compiler
    d. Multi-pass compiler

11. The regular expression have all strings of 0′s and 1′s with no two consecutive 0′s is :
    a. (0+1)
    b. (0+∈) (1+10)*
    c. (0+1)*
    d. (0+1)* 011

12. Is GCC a cross Complier
    a. Yes
    b. No

13. A compiler can check?
    a. Syntax Error
    b. Logical Error
    c. Both Logical and Syntax Error
    d. Not Logical and Syntax Error

14. Given the language L = {ab, aa, baa}, which of the following strings are in L*?
    1) abaabaaabaa
    2) aaaabaaaa
    3) baaaaabaaaab
    4) baaaaabaa
    a. 1, 2, 3
    b. 2, 3, 4
    c. 1, 3, 4
    d. 1, 2, 4

15. For every NFA a deterministic finite automaton (DFA) can be found that accepts the same language.
    a. True

b. False

16. Which one of the following options is true?
    a. The grammar in which every production at right hand side has only 1 alternative is always LL(1).
    b. Non-deterministic grammars are not LL(1).
    c. Left recursive & ambiguous grammar is not LL(1)
    d. All are true

17. When there is a reduce/reduce conflict?
    a. If a state does not know whether it will make a shift operation using the production rule i or j for a terminal
    b. If a state does not know whether it will make a shift or reduction operation using the production rule i or j for a terminal
    c. If a state e does not know whether it will make a reduction operation using the production rule i or j for a terminal
    d. None of the above

18. Number of elements in follow of A in the following grammar? T-> AB A-> a/b B-> c/d:
    a. 1
    b. 2
    c. 3
    d. 4

19. Which one of the following kinds of derivation is used by LR parsers?
    a. Rightmost in reverse
    b. Leftmost in reverse
    c. Leftmost
    d. Rightmost

20. Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?
    a. SLR, LALR
    b. CLR, LALR
    c. SLR, CLR
    d. LALR, CLR

21. Consider the augmented grammar given below :
    $S' \rightarrow S$, $S \rightarrow \langle L \rangle \mid id$, $L \rightarrow L,S \mid S$
    Let I_0 = CLOSURE ({[$S' \rightarrow \bullet S$]}). The number of items in the set GOTO (I_0 , ⟨) is: _____.
    a. 2
    b. 3
    c. 4
    d. 5

22. What is the maximum number of reduce moves that can be taken by a bottom-up parser for a grammar with no epsilon- and unit-production (i.e., of type A -> ϵ and A -> a) to parse a string with n tokens?
    a. n/2
    b. n-1
    c. n
    d. n+1

23. An LALR(1) parser for a grammar G can have shift-reduce (S-R) conflicts if and only if

a. The LR(1) parser for G has S-R conflicts.
b. The SLR parser for G has S-R conflicts
c. The LR(0) parser for G has S-R conflicts.
d. The LALR(1) parser for G has R-R conflicts.

24. Which of the following is a top down parser:
    a. Operator precedence parser
    b. Shift reduce parser
    c. Recursive descent parser
    d. LR(k) parsers

25. Consider the grammar with non-terminals N = {S,C,S1 },terminals T={a, b, i, t, e}, with S as the start symbol, and the following set of rules:
    S --> iCtSS1|a S1 --> eS|ϵ C --> b
    The grammar is NOT LL(1) because:
    a. Context free
    b. Ambiguous
    c. Left recursive
    d. Right recursive

26. A canonical set of items is given below
    S --> L. > R Q --> R. On input symbol < the set has
    a. A S-R and R-R conflict
    b. A S-R but not R-R conflict
    c. A R-R but not S-R conflict
    d. Neither S-R nor R-R conflict

27. 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}

28. Consider the grammar: S → (S) | a Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar be n1, n2 and n3 respectively. The following relationship holds good
    a. n1<n2<n3
    b. n1=n3<n2
    c. n1=n2=n3
    d. n1>n2>n3

29. The grammar S → aSa | bS | c is
    a. LL(1) but not LR(1)
    b. LR(1)but not LR(1)
    c. Both LL(1)and LR(1)
    d. Neither LL(1)nor LR(1)

30. Which of the following statements is false?
    a. An LL(1) parser is a top-down parser
    b. LALR is more powerful than SLR
    c. An ambiguous grammar can never be LR(k) for any k
    d. An unambiguous grammar has same leftmost and rightmost derivation

31. Consider the following source code:

```
c = a + b
d = c
c = c – e
a = d – e
b = b * e
b = d/b
```

a. No optimization is possible
b.
```
d = c
c = c – e
a = d – e
b = b * e
b = d/b
```
c.
```
c = a + b
d = c
c = c – e
a = d – e
b = d/b
```
d.
```
c = a + b
t = b * e
a = d – e
b = d/t
c = a
```

32. Peephole optimization is a form of
    a. Loop optimization
    b. Local optimization
    c. Data flow analysis
    d. Constant folding

33. Substitution of values for names (whose values are constants) is done in
    a. Loop optimization
    b. Local optimization
    c. Strength Reduction
    d. Constant folding

34. In compiler terminology reduction in strength means
    a. Replacing run time computation by compile time computation
    b. Removing loop invariant computation
    c. Removing common subexpressions
    d. Replacing a costly operation by a relatively cheaper one

35. Which of the following statements about peephole optimization is False?
    a. It is applied to a small part of the code
    b. It can be used to optimize intermediate code
    e. To get the best out of this, it has to be applied repeatedly
    f. It can be applied to the portion of the code that is not contiguous

36. The graph that shows basic blocks and their successor relationship is called

a. DAG
b. Control Graph
c. Flow Graph
d. Hamiltonian Graph

37. Dead-code elimination in machine code optimization refers to:
a. Removal of all labels
b. Removal of values that never get used
c. Removal of function which are not involved
d. Removal of a module after its use

38. Some code optimizations are carried out on the intermediate code because:
a. they enhance the portability of the compiler to other target processors
b. program analysis is more accurate on intermediate code than on machine code
c. the information from dataflow analysis cannot otherwise be used for optimization
d. the information from the front end cannot otherwise be used for optimization

39. DAG representation of a basic block allows
A Automatic detection of local common sub expressions
B Automatic detection of induction variables
C Automatic detection of loop variant
D None of the above

40. Which one of the following is FALSE?

a. A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end
b. Available expression analysis can be used for common subexpression elimination
c. Live variable analysis can be used for dead code elimination
d. x = 4 ∗ 5 => x = 20 is an example of common subexpression elimination

41. Compiler can check _____ error.
A. Logical
B. Syntax
C. Content
D. Both A and B

42. _____ is the most general phase structured grammar.
A. Context sensitive
B. Regular
C. Context free
D. All of these

43. When will the relationship between '+' and '-' be <
a) For unary minus
b) Minus is right associative
c) All of the mentioned
d) None of the mentioned

44. Which is the most powerful parser?
a) SLR
b) LALR
c) Canonical LR
d) Operator-precedence

45. Recursive descent parsing is an example of
a) Top down parsing
b) Bottom up parsing
c) Predictive parsing
d) None of the mentioned

46. How many minimum states are required to find whether a string has odd number of 0's or not?
a) 1
b) 2
c) 3
d) 4

47. Which loader function is accomplished by loader?
a) Reallocation
b) Allocation
c) Linking
d) Loading

48. Dividing a project into segments and smaller units in order to simplify design and programming efforts is called
a) Modular approach
b) Top down approach
c) Bottom up approach
d) Left right approach

49. Consider the following grammar:
S -> iCtSS1 | a
S1 -> eS | ε
C -> b
The grammar is NOT LL(1) because:
a) It is left recursive
b) It is right recursive
c) It is ambiguous
d) It is not context-free

50. 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}
c) {S -> FR} and {R -> * S}
d) {F -> id} and {R -> ε}

51. A grammar that produces more than one parse tree for some sentence is called _____
a)Ambiguous

b)Unambiguous
c)Regular
d) None of the mentioned

52.In Moore Machine O/P is associated with _____
a)Presentstate
b)Nextstate
c)Input
d) None of the mentioned

53. Which of the following features cannot be captured by CFG?
a) Syntax of if-then-else statements
b) Syntax of recursive procedures
c) A variable is declared before its use
d) Matching nested parenthesis

54. Semantic Analyser is used for?
a) Generating Object code
b) Maintaining symbol table
c) Generating Object code & Maintaining symbol table
d) None of the mentioned

55. When a computer is rebooted, a special type of loader is executed called?
a) Compile and GO loader
b) Boot loader
c) Bootstrap Loader
d) Relating Loader

56. Choose the correct statement.
a) CFG is not LR
b) Ambiguous Grammar can never be LR
c) CFG is not LR & Ambiguous Grammar can never be LR
d) None of the mentioned

57. Select a Machine Independent phase of the compiler.
a) Syntax Analysis
b) Intermediate Code generation
c) Lexical Analysis
d) All of the mentioned

58. A system program that combines the separately compiled modules of a program into a form suitable for execution?
a) Assembler
b) Compiler
c) Linking Loader
d) Interpreter

59 Type checking is normally done during _____
a) Lexical Analysis
b) Syntax Analysis
c) Syntax Directed Translation
d) Code generation

60 What does a Syntactic Analyser do?
a) Maintain Symbol Table

b) Collect type of information
c) Create parse tree
d) None of the above