Reg. No. : ☐☐☐☐☐☐☐☐☐☐☐☐

Question Paper Code:U5D02

B.E./B.Tech. DEGREE EXAMINATION, NOV 2024

Fifth Semester

Computer Science and Business Systems

21UCB502-COMPILER DESIGN

(Regulations 2021)

Duration: Three hours   Maximum: 100 Marks

PART A - (10 x 1 = 10 Marks)

1. _____ is considered as a sequence of characters in a token.   CO1- U

    (a) Texeme   (b) Pattern   (c) Lexeme   (d) Mexeme

2. How many tokens the statement sum=sum+I; has?   CO1- U

    (a) 5   (b) 6   (c) 10   (d) 15

3. Which of the following is left recursive production?   CO1- U

    i) E→E+T|T

    ii)T→U|T*U

    iii) E→id

    (a) i only   (b) ii only   (c) i and ii   (d) all of three

4. A grammar that produces more than one parse tree for some sentence is called as -----   CO1- U

    (a) Ambiguous   (b) Unambiguous   (c) Regular   (d) none

5. The phase 'Semantic Analysis' is responsible for ____ in Compiler.   CO1- U

    (a) Check semantics   (b) Static checking   (c)Type checking   (d) All of these

6. The representations of Intermediate code are   CO1- U

    i.Quadriple

    ii.Triple

    iii.Indirect Triple

    (a) i and ii   (b) i and iii   (c) ii and iii   (d) i, ii and iii

7. Whenever a procedure is executed, its activation record is stored on the stack, also known as?   CO1- U

    (a) Access Stack   (b) Control stack   (c) Formal Stack   (d) Return Stack

8. In Algebraic expression simplification, a = a + 1 can simply be replaced by?   CO1- U

    (a) a   (b) INC a   (c) DEC a   (d) MUL a

9. Optimization can be categorized broadly into ____ types.   CO1- U

    (a) 2   (b) 3   (c) 4   (d) 5

10. Before generating intermediate code, the compiler can modify the intermediate code by address calculations and improving loops   CO1- U

    (a) TRUE   (b) FALSE   (c) Can be true or false   (d) Can not say

<p align="center">PART – B (5 x 2= 10Marks)</p>

11. Define Translator and types of translators   CO1-U
12. Find FIRST and FOLLOW for the given grammar   CO2-App

    S→CC

    C→cC

    C→d

13. Define Three Address code.   CO1-U
14. Give the variety of outputs in the target programs.   CO1-U
15. Draw the DAG for the following basic block   CO1-App

    a= b + c

    b = a – d

    c = b + c

    d = a - d

<p align="center">PART – C (5 x 16= 80Marks)</p>

16. (a) Illustrate how the following high level language statement is transformed into machine code during the compilation process x=(a+b) * (c+d) with the neat sketch of phases of compiler.   CO2-App (16)

<p align="center">Or</p>

    (b) Illustrate the process of compilation for the program segment a=b+c*60 with a neat sketch.   CO2-App (16)

17. (a) Design Predictive parser for the Grammar CO2-App (16)

S->(L) |a

L->L,S|S

and parse the input string (a, a)

Or

(b) Design the predictive parser for the following grammar and parse CO2-App (16)
the string and check whether the string ((id+id)*id) is valid or not.

E→E+T |T

T→T*F |F

F →(E) |id

18. (a) Describe the role of syntax-directed translation in handling Boolean CO1-U (16)
expressions within a compiler. Provide a detailed explanation of the
translation scheme, including the use of attributes and rules.

Or

(b) Explain the translation of Arithmetic expressions in detail. CO1-U (16)

19. (a) Construct the DAG for the following basic block CO2-App (16)

1. t1:=4*i

2. t2:=a[t1]

3. t3:=4*i

4. t4:=b[t3]

5. t5:=t2*t4

6. t6:=prod+t5

7. prod:=t6

8. t7:=i+1

9. i:=t7

10. if i<=20 goto 1

Or

U5D02

(b) Convert the following three address code into flow graph and optimze the flow graph

CO2-App (16)

1. i=1
2. j=1
3. t1=10*i
4. t2=t1+j
5. t3=8*t2
6. t4=t3-88
7. a[t4]=0
8. j=j+1
9. if j<=10goto 3
10. i=i+1
11. if i<=10 goto 2
12. i=1
13. t5=i-1
14. t6=88*t5
15. a[t6]=1
16. i=i+1
17. if i<=10 goto 13

20. (a) Describe in detail about Heap Management. CO1-U (16)

Or

(b) Explain the data access without nested procedure and the issues with nested procedure. CO1-U (16)