Daffodil International University

Faculty of Science & Information Technology

Department of Computer Science and Engineering

Final Examination, Spring 2024

Course Code: CSE 331 , Course Title: Compiler Design

Level: 4 Term: 1 Batch: 59

Time: 2:00 Hrs **Total Marks: 40**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1.		**Expr -> Expr + Term | Term** **Term -> Term * Factor | Factor** **Factor -> (Expr) | id**		CO2
	a)	**Convert** the given left-recursive grammar into right-recursive grammar.	[3]	
	b)	**Find** FISRT() and FOLLOW() for the right-recursive grammar.	[3]	
	c)	**Check** whether the right-recursive grammar is LL(1) parser or not	[4]	
2.	*a)*	**Produce** Canonical Table from the following grammar S -> L = R L -> * R | id R -> L	[6]	CO2
	b)	Suppose that the input string is id = * id. **Find** whether the input string is accepted or rejected for the given (2a) **LR(0)** Parser.	[4]	
3.	*a)*	**Convert** the following arithmetic expression into various intermediate representations: **((rate*(vat+tax)) - ((cost*tax)+(revenue+(vat+tax)))** a) Represent the expression using Three Address Code. b) Represent the expression using Quadruples. c) Implement a Triples data structure to represent the expression. d) Construct the syntax tree for the expression. e) Create a Directed Acyclic Graph (DAG) for the expression.	[10]	CO3

4.	a)	Consider the following three address codes and answer the questions.	[10]	CO3

```
1.  x := 5
2.  y := 10
3.  z := x + y
4.  t1 := z - 2
5.  x := t1 + y
6.  if x < y goto 7
7.  t2 := x * y
8.  y := 15
9.  z := z + y
10. w := x - y
11. if t1 = 0 goto 17
12. w := x + y
13. t3 := y - z
14. z := w - t3
15. if z < w goto 3
16. z := x + y
17. output z
18. output w
19. x := 10
20. y := 15
21. a := x * y
22. t1 := a + 3
23. x := t1 + y
24. if x < y goto 25
25. t2 := x + y
26. y := 15
27. a := a + y
28. b := x - y
29. if t1 = 0 goto 35
30. b := x * y
31. t3 := y - z
32. a := b - t2
33. if b < b goto 21
34. a := x + y
35. output a
36. output b
```

a) **Identify** the lines of code that serve as leader instructions according to leader selection rule 2.

b) **Identify** the lines of code that serve as leader instructions according to leader selection rule 3.

c) **Identify** the lines which are selected as leader instruction according to both selection rule 2 and 3.

d) **Draw** the control flow graph (CFG) for the given three-address code.

e) **Explain** how the Constant Propagation optimization technique could be implemented on this code, mentioning the lines where this optimization can be applied.