Reg. No.								

Question Paper Code

13450

B.E. / B.Tech. - DEGREE EXAMINATIONS, APRIL / MAY 2025

Sixth Semester

Computer Science and Engineering

(Common to Computer Science and Engineering (IoT))

20CSPC602 - COMPILER DESIGN

Regulations - 2020

	Regulations -	- 2020							
Duration: 3 Hours Max.									
$PART - A (MCQ) (10 \times 1 = 10 Marks)$									
	Answer ALL Qu	estions		Marks	Level	CO			
1.	Which computer program accepts the high-level lan	guage and conver	ts it into assembly	1	K1	CO1			
	language?		•						
		sembler	(d) Compiler						
2.	How many phases are there in the Compilation proces		. , 1	1	K1	CO1			
	(a) 7 (b) 6 (c) 8		(d) 4						
3.	In the context of lexical analysis, which of the followi	ng is an example of	` '	1	K1	CO2			
		-) All of the above						
4.	Which of the following is true about lexical analyzers	,	<i>'</i>	1	<i>K1</i>	CO2			
	(a) They stop execution immediately when an error is								
	(b) They report errors but continue analyzing the rest								
	(c) They automatically fix the errors encountered								
	(d) They do not need to report errors								
5.	Show the main difference between LL and LR parsers	.		1	<i>K</i> 2	CO3			
	(a) LL parsers are top-down, while LR parsers are bot								
	(b) LL parsers use more memory than LR parsers.	voiii wp.							
	(c) LR parsers cannot handle left-recursive grammar.								
	(d) LL parsers cannot handle right-recursive grammar								
6.	Which of the following is NOT a type of Syntax Anal			1	K1	CO3			
٠.	(a) Top-down parser (b) Bottom-up parser (c) Recurs	₹	(d) Tokenizer						
7.	Which of the following is a common form of intermed	_		1	K1	CO4			
, .	modern compilers?	nate representation	r asca in many						
		b) Abstract Syntax	Tree (AST)						
		d) Assembly code	1100 (1151)						
8.	Which of the following is typically done during intern	,	ation ⁹	1	K1	CO4			
٥.	(a) Lexical analysis	_	assembly code						
	(c) Performing machine-independent optimizations	(d) Error handl	•						
9.	What is register allocation in the context of code gene	` '	····S	1	K1	CO5			
	(a) Assigning variables to registers to minimize memory								
	(b) Generating machine code from the intermediate co	•							
	(c) Selecting instructions for the target architecture	, 60							
	(d) Mapping syntax tree nodes to instructions								
10.	What does the "constant folding" optimization do?			1	K1	CO6			
	(a) Replaces a function call with a constant value								
	(b) Moves constant expressions out of loops								
	(c) Evaluates constant expressions at compile-time and	d replaces them wi	th their values						
	(d) Combines multiple loops into a single loop	o reproces unem wa	, W.						
	(a) comomes muniple roops mio a single roop								
$PART - B (12 \times 2 = 24 Marks)$									
	Answer ALL Questi	•							
11.	List the various Compiler Construction tools.			2	<i>K1</i>	CO1			
	-	n 1 - 86 6			124	50			
K1 -	- Remember; K2 – Understand; K3 – Apply; K4 – Analyze; K5 – I	zvaluate; Ko – Create			134.	5 <i>U</i>			
	I								

13. Interpret the transition diagram for an identifier. 14. List the possible error recovery actions in lexical analyzer. 15. Convert Non Deterministic Grammar to Deterministic Grammar S->iEtS iEtSeS a E->b 16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is toode motion? 22. K1 C05 23. What is the induction variable? PART - C (6 × 11 = 66 Marks) Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. 5 K2 C01 OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression 11 K2 C02 (a b)*abb. OR						
13. Interpret the transition diagram for an identifier. 14. List the possible error recovery actions in lexical analyzer. 15. Convert Non Deterministic Grammar to Deterministic Grammar S-S-IETS IETSeS a E->b 16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. What is the induction variable? 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. 24. (ii) Discuss the cousins of the compiler. OR 25. Explain the various phases of compiler and trace it with the program segment c=a+b*5. 26. (ii) Describe in detail the tool for generating lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail the tool for generating lexical analyzer with an example program. (iv) Describe in detail the tool for generating lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyses. 10. Co2 21. KI Co3 22. KI Co4 23. CO3 24. CO5 25. a) Construct LL(1) parsing table for the following grammar 25. SiEtS iEtSeS a 26. CO2 27. CO3 28. CO3 29. CO3 20. CO3 20. CO3 20. CO3 20. CO3 21. CO4 21. CO5 22. KI CO4 23. CO5 24. CO5 25. CO4 26. CO2 27. CO5 28. CO3 29. CO4 20. CO5 20. CO5 20. CO5 20. CO5 20. CO5 21. CO5 20. CO5 21. CO5 21. CO5 22. CO5 23. CO5 24. CO5 25. CO5 26. CO5 27. CO5 28. CO5 29. CO5 20. CO5 20. CO5 20. CO5	12.	Illustra	ate diagrammatically how a language is processed.	2	<i>K</i> 2	CO1
14. List the possible error recovery actions in lexical analyzer. 15. Convert Non Deterministic Grammar to Deterministic Grammar S->iEtS iEtSeS a E->b 16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. K1 Cod 23. What is the induction variable? PART - C (6 × 11 = 66 Marks) Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. 5 K2 Col OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 11. K3 Co3 S->iEtS iEtSeS a E->b				2	<i>K</i> 2	CO2
15. Convert Non Deterministic Grammar to Deterministic Grammar S.>iEtS iEtSeS a E->b 16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. K1 Code and the induction variable? 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. 5 K2 Cole and Translate DFA using Direct method for the following Regular Expression (ii) Describe in detail the tool for generating lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical analyzer with an example program. (iv) Describe in detail about the issues in lexical a				2	K1	CO2
S->iEtS iEtSeS a E->b 16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is register descriptor and address descriptor? 22. What is the induction variable? 23. What is the induction variable? 24. I Co6 25. What is the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. 26. (ii) Discuss the cousins of the compiler. 27. What is the various phases of compiler and trace it with the program segment c=a+b*5. 28. Translate DFA using Direct method for the following Regular Expression 11 K2 CO2 (a b)*abb. OR 29. b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 10. Co3 11. K3 CO3 12. CO3 12. CO3 13. CO3 14. CO3 15. S->iEtS iEtSeS a E->b			· · · · · · · · · · · · · · · · · · ·	2	<i>K3</i>	CO3
E->b 16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. KI COS 23. What is the induction variable? 23. A) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. (iii) Discuss the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression for COS (iii) Describe in detail the tool for generating lexical analyzer with an example program. (iii) Describe in detail about the issues in lexical analysis. (iv) COS (value)*	10.					
16. What is an Operator Precedence Parser? 17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. K1 Code and the induction variable? 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. 23. a) (ii) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. 24. a) Translate DFA using Direct method for the following Regular Expression call b) (i) Describe in detail the tool for generating lexical analyzer with an example program. 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b						
17. What is static checking? 18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. What is the induction variable? 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. 10. Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression for Delimiters and keywords and construct transition (a b)*abb. 11. K2 CO2 12. What is the induction variable? 12. K1 CO3 13. (ii) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (iii) Discuss the cousins of the compiler. 5. K2 CO1 11. K2 CO2 12. (a b)*abb. 12. (a b)*abb. 13. (a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. 14. (a) Translate DFA using Direct method for the following Regular Expression (a) Explain the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 17. (a) CO3 18. (b) (b) CO4 19. (c) CO4 19. (c) CO5 10. (c) CO6 10. (c) CO6 11. (c) CO6 12. (c) CO6 13. (c) CO6 14. (c) CO6 15. (c) CO6 16. (c) CO6 17. (c) CO6 18. (c) CO6 19. (c) CO6 10. (c) CO6 11. (c) CO6 11. (c) CO6 12. (c) CO6 13. (c) CO6 14. (c) CO6 15. (c) CO6 16. (c) CO6 17. (c) CO6 18. (c) CO6 19. (c) CO6 19. (c) CO6 10. (c) CO6 10. (c) CO6 11. (c) CO6 11. (c) CO6 12. (c) CO6 13. (c) CO6 14. (c) CO6 15. (c) CO6 16. (c) CO6 17. (c) CO6 18. (c) CO6 19. (c) CO6 10. (c) CO6 11. (c) CO6 11. (c) CO6 12. (c) CO6 13. (c) CO6 14. (c) CO6 15. (c) CO6 16. (c) CO6 17. (c) CO6 18. (c) CO6 19. (c) CO6 10. (c) CO6 10. (c) CO6 11. (c) CO6 11. (c) CO6 12. (c) CO6 13. (c) CO6 14. (c) CO6 15. (c) CO6 16. (c) CO6 17. (c) CO6 18. (c) CO6 19. (c) CO6 19. (c) CO6 10. (c) CO6 10. (c) CO6 10. (c) CO6 11.	16		is an Operator Precedence Parser?	2	K1	CO3
18. Define Backpatching. 19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. What is the induction variable? 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b			<u> </u>	2		
19. What is an Activation tree? 20. What is register descriptor and address descriptor? 21. What is code motion? 22. What is the induction variable? PART - C (6 × 11 = 66 Marks) Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b			· · · · · · · · · · · · · · · · · · ·			
20. What is register descriptor and address descriptor? 21. What is code motion? 22. What is the induction variable? 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression [11] K2 COI (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b				2		
21. What is code motion? 22. What is the induction variable? PART - C (6 × 11 = 66 Marks) Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 2 K1 C06 K2 C01 K3 C01 K4 C02 K5 C02 C02 C03 C04 C05 C05 C06 C07 C07 C08 C09 C09 C09 C09 C09 C09 C09						
PART - C (6 × 11 = 66 Marks) Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b						
PART - C (6 × 11 = 66 Marks) Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b						
Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b	22.	wnat	is the induction variable?	2	KI	COO
Answer ALL Questions 23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b			$PART - C (6 \times 11 - 66 Marks)$			
23. a) (i) Write the Regular Expression for Delimiters and keywords and construct transition diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. Cos 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b						
diagrams for the same. (ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b	23	a) (i)		6	K2	CO1
(ii) Discuss the cousins of the compiler. OR b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b	23.	a) (1)	· · · · · · · · · · · · · · · · · · ·			
b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b		(ii)		5	K2	CO1
b) Explain the various phases of compiler and trace it with the program segment c=a+b*5. 24. a) Translate DFA using Direct method for the following Regular Expression of the following segment of the following forms of the following forms of the following forms of the following forms of the following grammar of the following grammar of the following grammar of the following forms of the following grammar of the following gramm		(11)	<u>•</u>			001
24. a) Translate DFA using Direct method for the following Regular Expression 11 K2 CO2 (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. 6 K2 CO2 (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b		b)		11	K2	CO1
24. a) Translate DFA using Direct method for the following Regular Expression 11 K2 CO2 (a b)*abb. OR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. 6 K2 CO2 (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b		U)				001
oR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b			C-a+0 3.			
oR b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b	24	2)	Translate DEA using Direct method for the following Pagular Expression	11	K2	CO2
b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b	∠4.	a)				
b) (i) Describe in detail the tool for generating lexical analyzer with an example program. (ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b						
(ii) Describe in detail about the issues in lexical analysis. 5 K2 CO2 25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b		b) (i)		6	K2	CO2
25. a) Construct LL(1) parsing table for the following grammar S->iEtS iEtSeS a E->b						
S->iEtS iEtSeS a E->b		(11)	Describe in detail about the issues in lexical analysis.	J	K2	CO2
S->iEtS iEtSeS a E->b	25	`		11	V2	CO3
E->b	25.	a)		11	K3	003
2,0						
Also check whether the given grammar is LL(1) or not?			2.0			
O.D.						
OR		1 \		11	V2	CO2
1) G GLD		b)		11	K3	COS
b) Construct a SLR parsing table by examining the following grammar.						
S->CC			C->cC d and use the string "cdd" to parse with the SLR Parsing Table.			
b) Construct a BER parsing table by examining the following grammar.	26	,		11	V2	CO1
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table.	26.	a)		11	KJ	CO4
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment 11 K3 CO4						
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b.		1.		11	v2	CO1
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR		b)		11	K3	C <i>04</i>
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). 11 K3 CO4			(1) Syntax tree (11) Quadruples (111) Triples (11) Indirect Triples			
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR	27	\			W)	COF
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples	27.					
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5		(ii)		5	K2	CO5
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5 (ii) Explain any four issues when designing a code generator.			OR			
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5 (ii) Explain any four issues when designing a code generator. OR		b)	Explain about runtime storage management.	11	<i>K</i> 2	CO5
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5 (ii) Explain any four issues when designing a code generator. 5 K2 CO5						
OR		b)	Construct a SLR parsing table by examining the following grammar. S->CC		11	11 K3
1) G GLD		b)		11	K3	COS
b) Construct a BER parsing table by examining the following grammar.						
S->CC			C->cC d and use the string "cdd" to parse with the SLR Parsing Table.			
S->CC						
S->CC	26.	a)	Generate an intermediate code for the following code segment	11	<i>K3</i>	CO4
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table.	20.	u)				
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment 11 K3 CO4						
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b.		b)		11	K3	CO4
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR		U)		11	113	007
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). 11 K3 CO4			(1) Syntax tree (11) Quadruples (111) Triples (11) Indirect Triples			
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). 11 K3 CO4						
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). 11 K3 CO4	27	a) (i)	Explain about simple code generation algorithm in detail	6	<i>K</i> 2	CO5
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples				_	W2	CO5
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5		(11)	Explain any four issues when designing a code generator.	3	KΖ	COS
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5			OR			
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5 (ii) Explain any four issues when designing a code generator.		1 \		11	νn	CO5
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5 (ii) Explain any four issues when designing a code generator. 5 K2 CO5		b)	Explain about runtime storage management.	11	NΔ	003
S->CC C->cC d and use the string "cdd" to parse with the SLR Parsing Table. 26. a) Generate an intermediate code for the following code segment if (a > b) x = a + b else x = a - b. OR b) Construct the following using the arithmetic expression a+- (b+c). (i)Syntax tree (ii)Quadruples (iii)Triples (iv)Indirect Triples 27. a) (i) Explain about simple code generation algorithm in detail. 6 K2 CO5 (ii) Explain any four issues when designing a code generator. 5 K2 CO5						

- 28. a) Explain in detail about optimization of Basic Blocks and Construct the DAG for the following Basic Block.
 - 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

d₇: i := u3

EXIT

b) Apply global data flow analysis for the following.

B3

 d_6 : a := u2

B1 d₁: i := m-1 d₂: j := n d₃: a := u1

B2 d₄: i := i+1 d₅: j := j - 1

11 K3 CO6

11

K3 CO6