

- N. B.:** (1) Attempt any **three** questions from **each section**.  
 (2) **Figures** to the right indicate full marks.  
 (3) Assume additional data if necessary but state the same clearly.  
 (4) Symbols have their usual meanings and tables have their usual standard design unless stated otherwise.

### Section I

- |   |    |  |           |
|---|----|--|-----------|
| 1 | a. | Consider a string $a=b*(c-d)+(c*d-100)$<br>Show the steps which will be followed by compiler to process the statement in each phase.   | <b>07</b> |
|   | b. | Explain the concept of LEX with respect to following points:<br>Definition, syntax and example.  | <b>06</b> |
| 2 | a. | How to minimize DFA? Explain appropriate algorithm with suitable example.  | <b>05</b> |
|   | b. | Draw DFA for accepting a valid identifier name. Also mention regular expression for the same.  | <b>05</b> |
|   | c. | Explain the representation of an array in two dimensional using Row major form.  | <b>03</b> |
| 3 | a. | Compute First and Follow for the following grammars:<br>1) $S \rightarrow aB/bA$<br>$A \rightarrow bAA/aS/a$<br>$B \rightarrow aBB/bS/b$<br>2) $S \rightarrow 0A1/0A0$<br>$A \rightarrow 0A0/1$<br>3) $S \rightarrow A*B/*A$<br>$A \rightarrow \#B/B\#$<br>$B \rightarrow *A/\#$ | <b>06</b> |
|   | b. | What is the difference between static storage and dynamic storage allocation?  | <b>04</b> |
|   | c. | Construct CFG for a language over $\{0, 1\}$ which accepts a string starting with 0 and ending with 1.   | <b>03</b> |
| 4 | a. | Consider the following grammar and input string. Parse the string using bottom up parsing technique using the concept of stack.<br>$S \rightarrow S+C/B$<br>$B \rightarrow B\&C/C$<br>$C \rightarrow *C/a/b$<br>String = $*a\&*b$  | <b>06</b> |
|   | b. | Discuss the concept of predictive parsing with suitable diagram. Also mention when the grammar is considered as LL(1)?   | <b>05</b> |
|   | c. | State the advantages of LR parsers.  | <b>02</b> |

5 a. Consider the following Production Rules and Parsing Table:

06

1.  $S' \rightarrow S$
2.  $S \rightarrow aABe$
3.  $A \rightarrow Abc$
4.  $A \rightarrow b$
5.  $B \rightarrow d$

state	Action						GOTO		
	a	b	c	d	e	\$	S	A	B
0	s2						1		
1						acc			
2		s4						3	
3		s6		s7					5
4		r4		r4					
5					s8				
6			s9						
7					r5				
8						r2			
9		r3		r3					

For the string “**abbcbcde\$**” show LR(0) parsing moves.

b. Check whether following Grammar is in SLR (1) or not?

06

- $S \rightarrow CC$   
 $C \rightarrow cC/d$

\*\*\*\*

### Section II

6 a. What is a syntax direct translation? Explain its types with suitable examples.

04

b. Write syntax direct translations for the following Grammar. The Production

04

Rules are as follows:

- $S \rightarrow E\$$   
 $E \rightarrow E+E$   
 $E \rightarrow E * E$   
 $E \rightarrow I$   
 $I \rightarrow I \text{ digit}$   
 $I \rightarrow \text{digit}$

c. Convert the given infix expression into postfix:

04

- a)  $a*(b-c)+d*e$
- b)  $a+(b*c)/(-b*-c+d)$

- 7 a. Explain the format of symbol table by giving suitable example. **04**  
b. Discuss the format of activation records? **04**  
c. Write translation rules for following: **04**  
Type →struct {fieldlist}  
    | ptr  
    |char  
    | int  
    | float  
    | double  
Fieldlist →fieldlist field ;  
          | field  
Field →type id | field {integer}]
- 8 a. Explain machine independent optimization technique. **04**  
b. State the need of DAG and explain the concept with suitable example. **04**  
c. Convert the following code into flow graph: **04**  
    1) C=1  
    2) Goto 5  
    3) M=1  
    4) C=B\*3  
    5) B=B+1  
    6) If C <=20 goto 3
- 9 a. Define dominators and mention properties of the same. **03**  
b. What is directed acyclic graph?(DAG)? Explain algorithm for constructing DAG. **05**  
c. Discuss loop unrolling and loop jamming techniques. **04**
- 10 a. Explain different types of addressing modes. **06**  
b. Consider the following Three-Address code statements and write possible code sequence in assemble level language. [Assume suitable machine architecture ] **06**  
    T=A-B  
    U=A-C  
    V=T+U  
    W=V+U

\*\*\*\*