

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
TEST -3 EXAMINATIONS- 2026
B.Tech-VI Semester (CSE)

COURSE CODE (CREDITS): 18B11CI612 (3)

MAX MARKS: 35

COURSE NAME: COMPILER DESIGN

COURSE INSTRUCTOR: Pardeep, Ramesh, Seema, Nitika

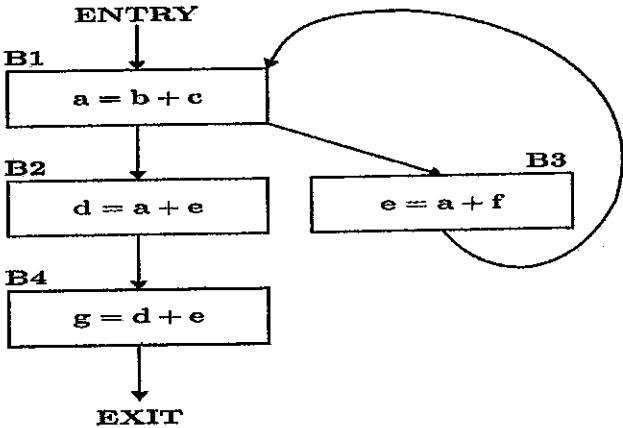
MAX. TIME: 2 Hours

Note: (a) All questions are compulsory.

(b) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

(c) Use of calculator is not allowed

Q.No	Question	CO	Marks
1	Consider the following Syntax Directed Translation and execute the arithmetic expression $8+4*(6-2)/2$ using bottom up and top down parsing $E \rightarrow E1 + T \{E.val = E1.val + T.val\}$ $E \rightarrow E1 - T \{E.val = E1.val - T.val\}$ $T \rightarrow T1 * F \{T.val = T1.val * F.val\}$ $T \rightarrow T1 / F \{T.val = T1.val / F.val\}$ $T \rightarrow F \{T.val = F.val\}$ $F \rightarrow (E) \{F.val = E.val\}$ $F \rightarrow \text{digit} \{F.val = \text{digit.lexval}\}$	6	3.5+3.5
2	(i) Consider the following grammar $T \rightarrow T+T/T*T/id$ and construct the operator precedence relation table. Show the step by step processing of the input string $id+id*id$ using your constructed precedence table. (ii) Make the operator function table and the graph for $T \rightarrow T+T/T*T/id$ and explain the precedence of operators while execution for your constructed	5	4+4
3	Consider the given 3-address code sequence. The code sequence is to be split by into basic blocks by the underlying compiler. 1. $i = 1$ 2. $\text{total} = 0$ 3. if $i \leq 5$ goto 18 4. $j = 1$ 5. if $j > 4$ goto 15 6. $t1 = i * j$ 7. $t2 = t1 + \text{total}$ 8. $\text{total} = t2$ 9. $t3 = i + j$ 10. $t4 = t3 * 2$ 11. $\text{temp} = t4$ 12. $j = j + 1$ 13. goto 5 14. printf("Hello"); 15. $i = i + 1$ 16. goto 3	7	7

	17. printf("Hello"); 18. print total 19. end Show the control flow graph to identify loops. How many numbers of basic building blocks are there and which basic building block is the largest one in terms of number of instructions?		
4	Consider the following three-address code: 1. $i = 1$ 2. while($i \leq 20$) 3. { 4. $t1 = a * b$ 5. $t2 = c + d$ 6. $t3 = i * 4$ 7. $t4 = a * b$ 8. $t5 = t4 + t2$ 9. $A[t3] = t5$ 10. $t6 = i * 4$ 11. $B[t6] = t1 + t2$ 12. $t7 = m + n$ 13. $x = t7 * i$ 14. $i = i + 1$ 15. } 16. print A[4] a. Detect the following in the above given 3 address code: (i) loop invariants (ii) common subexpressions b. Apply the following (i) code motion (ii) strength reduction (iii) common subexpression elimination and generate optimized code as per part (b).	6	2+6
5	 <pre> graph TD ENTRY --> B1 B1["B1 a = b + c"] --> B2 B2["B2 d = a + e"] --> B4 B4["B4 g = d + e"] --> EXIT B1 -- loop --> B3 B3["B3 e = a + f"] --> B1 EXIT["EXIT"] </pre> Consider the above control flow graph and do the liveness analysis for all blocks.	8	5