



VIT

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING CONTINUOUS ASSESSMENT TEST - II FALL SEMESTER 2024-2025

REG.NO.:

SLOT: C1+TC1

Programme Name & Branch : B.Tech CSE
Course Code and Course Name : BCSE307L Compiler Design
Faculty Name(s) : PROF. SAHAAYA ARUL MARY S A, PROF. KANNADASAN R, PROF. VISHNUPRIYA, PROF. VETRISSELVI T, PROF. BHUVANESWARI M, PROF. KANAGARAJ R, PROF. KALAIVANI K, PROF. SATHYA K, PROF. BASKARAN P, PROF. SABYASACHI KAMILA, PROF. UMA PRIYA D, PROF. MUKKU NISANTH KARTHEEK, PROF. ARUMUGA ARUN R, PROF. ISLABUDEEN M, PROF. SUGANTHINI C, PROF. NAGA PRIYADARSINI R, PROF. BHAWANA TYAGI, PROF. DEBI PRASANNA ACHARJYA, PROF. BAIJU B V, PROF. UMAMAHESWARI M

Class Number(s) : VL2024250101542, VL2024250101548, VL2024250101555,
 VL2024250101587, VL2024250101605, VL2024250101612, VL2024250101623, VL2024250101633,
 VL2024250101641, VL2024250101651, VL2024250101660, VL2024250101669, VL2024250101673,
 VL2024250101676, VL2024250101684, VL2024250101725, VL2024250101734, VL2024250101673,
 VL2024250101746, VL2024250107999 VL2024250101740

Date of Examination : 15-Oct-2024

Exam Duration : 90 minutes

Maximum Marks: 50

General instruction(s):

CO2: Develop language specifications using context free grammars (CFG)

CO3: Apply the ideas, the techniques, and the knowledge acquired for the purpose of developing software systems.

CO4: Constructing symbol tables and generating intermediate code.

CO5: Obtain insights on compiler optimization and code generation

Q. No	Question	M	CO	BL
1.	<p>The grammar given below is used to parse all possible expressions in C that involve(only) variables of pointer datatype(ignoring the semi-colon).</p> <p><i>program</i> → <i>expression</i> <i>assignment</i> <i>assignment</i> → <i>expression</i> = <i>expression</i> <i>expression</i> → <i>identifier</i> *<i>expression</i> (Assume that identifier is a terminal symbol)</p> <p>Check whether the above grammar is SLR(1). If yes, then parse the following: <i>*ptr1 = ptr2</i></p>	10	CO2	BL4
2.	<p>Consider the following grammar for Boolean expressions and if-else statements:</p> <p><i>S</i> → <i>if B then S1 else S2</i> <i>S</i> → <i>assign id = E</i> <i>B</i> → <i>E == E</i> <i>E > E</i> <i>E</i> → <i>E + T</i> <i>E - T</i> <i>T</i> <i>T</i> → <i>T * F</i> <i>T / F</i> <i>F</i> <i>F</i> → <i>(E)</i> <i>num</i></p> <p>Write a SDD that evaluates arithmetic expressions(E,T,F). Evaluates Boolean expressions(B) . Given the below input, construct the annotated parse tree showing how your SDD evaluates the Boolean expression and simulates the conditional assignment: <i>if (3 + 4) > 5 then assign x = 10 else assign x = 20</i></p>	10	CO3	BL4



VIT

Vellore Institute of Technology
(Deemed to be University under section 3 of IGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
FALL SEMESTER 2024-2025

REG.NO.:

SLOT: C1+TC1

3.	<p>Consider the following C like code that involves pointers and dynamic memory allocation.</p> <pre> int* allocateAndFillArray(int size) { int* arr = (int*) malloc(size * sizeof(int)); for (int i = 0; i < size; i++) { arr[i] = i * 2; } return arr; } int main() { int n = 5; int* p = allocateAndFillArray(n); int sum = 0; for (int i = 0; i < n; i++) { sum += p[i]; } return sum; } </pre> <p>Generate the three-address code(TAC) for the above code. Assume that the TAC has an instruction alloc(n) that performs the malloc().</p>	10	CO4	BL5
4.	<p>Consider the following C program, that transposes a 3 x 3 matrix and sums the diagonal elements:</p> <pre> for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) { transpose[j][i] = arr[i][j]; if (i == j) { sum = sum + transpose[i][j]; } } } </pre> <p>Convert the above code into three-address code and partition the same into basic blocks. Construct the flow graph for the basic blocks, showing control flow between them.</p>	10	CO5	BL5



VIT

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
FALL SEMESTER 2024-2025

REG.NO.:

SLOT: C1+TC1

5.	<p>a. Consider the following C code, that uses a switch-case statement to compute different operations based on a input value. (5 Marks)</p> <pre> int result; int x = 5; switch(x) { case 1: result = x + 1; break; case 2: result = x * 2; break; case 3: result = x - 3; break; default: result = x; } </pre> <p>Write semantic actions to generate intermediate code for the above switch-case statement. Show the annotated parse tree for the switch case statement.</p>	10	CO4 & CO5	BL4
	<p>b. Consider the assembly code generated by a compiler. Apply peephole optimization to optimize the code. (5 Marks)</p> <ol style="list-style-type: none"> 1. MOV R1, R2 2. MOV R2, R1 3. ADD R3, #0 4. JMP L1 5. MOV R4, #5 6. L1: MUL R5, #1 7. MOV R6, #0 8. ADD R6, R7 9. CMP R8, R9 10. JNE L2 11. JMP L3 12. L2: JMP L3 13. L3: NOP 			