

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

|  |  |
| --- | --- |
| **Course code: CSA14** | **Course Name: Compiler Design** |
| **Branch : CSE** | **Academic year: 2025- 2026** |
| **Date of Exam:** **13-08-2025** | **Max.Marks: 30** |

**ANSWER ALL THE QUESTIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **CLASS TEST - IV** | **Marks** | **CO** | **BTL** | **PO** |
| 1 | Construct three address code using the following production for Boolean expression (a < b) and (c < d) or (e < f)  B → B1 || B2  B → B1 && B2  B → ! B1  B → E1 relop E2  B → true  B → false  E → id | 10 | CO4 | 3 | PO1  ’  PO3 |
| 2 | Given the following Boolean expression: ((a AND b) OR (c AND (NOT d))) AND (e OR (f XOR g))  a) Convert the expression into its equivalent intermediate code using three-address code representation.  b) Optimize the intermediate code generated, if possible, by minimizing the number of temporary variables used. | 10 | CO4 | 3 | PO1  ’  PO3 |
| 3 | Consider the following code segment written in a high-level programming  language:  ……  cpp  if (a > b) {  x = y + z;  } else {  x = y - z;  Using the backpatching technique for intermediate code generation, perform the following tasks:  a) Generate the intermediate code for the above code segment using a three-address code format.  b) Show the process of backpatching the jump statements to handle the 'if and `else' conditions correctly.  Assume that the following symbols and addresses are used:  'a', 'b', 'x', 'y', and 'z' are all variables available in memory. Intermediate labels are represented as 'LI', 'L2', `L3', etc. | 10 | CO4 | 3 | PO1  ’  PO3 |