

Time: 3 hours

Max. Marks: 75

Note: This question paper contains two parts A and B.  
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

## PART - A

(25 Marks)

- Write a brief note on bootstrap process [2]  
 What are the differences between a compiler and an interpreter? [3]  
 Give the specification of the YACC parser generator [2]  
 Construct the LR(0) items for the "dangling-else" grammar. [3]  
 How to check structural equivalence of two type expressions? [2]  
 Define and write the differences between synthesized attributes and inherited attributes. [3]  
 Write a short note on Flow graph. [2]  
 Write an algorithm for constructing a basic block. [3]  
 Define various possible outputs of the code generator [2]  
 Construct DAG for the following basic block: [3]  
 $T1=A+B$   
 $T2=C+D$   
 $T3=E - T2$   
 $T4=T1- T3$

## PART - B

(50 Marks)

- 2.a) Explain various error recovery strategies in lexical analysis  
 b) Construct a Finite Automata and Scanning algorithm for recognizing identifiers, numerical constants in C language [5+5]  
 OR  
 3) Explain the various phases of a compiler with an illustrative example [10]  
 4. Construct the LR Parsing table for the following grammar [10]  
 $E \rightarrow E + T \mid T$   
 $T \rightarrow T * F \mid F$   
 $F \rightarrow (E) \mid id$   
 OR  
 5.a) Write a YACC program that will take regular expression as input and produce its parse tree as output  
 b) Write an algorithm for computing LR(k) item-sets. [5+5]

- 6.a) Write an SDT to convert infix to postfix expression.  
 b) Explain briefly about polymorphic functions. [5+5]  
 OR

7. Explain various storage allocation strategies with its merits and demerits. [10]

8) Discuss various techniques of function preserving transformations for code optimization. [10]

9. Explain how data flow equations are set up and solved for improving code. [10]

10. Explain the following peephole optimization techniques [5+5]

- a) Elimination of Redundant Code  
 b) Elimination of Unreachable Code.

11) Explain in detail about machine dependent code optimization techniques with their drawbacks. [10]