Hall Ticket Number:

Code No. : 32511

VASAVI COLLEGE OF ENGINEERING (Autonomous), HYDERABAD
B.E. (I.T.) III Year II-Semester Main Examinations, May-2017

Compiler Construction

Time: 3 hours

Max. Marks: 70

Note: Answer ALL questions in Part-A and any FIVE from Part-B

Part-A (10 × 2 = 20 Marks)

1. Define Lexical error. Give an example.
2. Write a regular expression to identify floating point numbers.
3. When is it useful to eliminate left-recursion from a grammar and why?
4. What is Shift Reduce Parsing?
5. List the applications of Syntax Directed Definition (SDD).
6. List the various forms of three address instructions.
7. Write the design goals of garbage collector.
8. Define Flow Graphs.
9. Define instruction cost. Determine the cost of the following instruction sequences.

   LD R_0, i

   MUL R_0, R_0, #8

   LD R_1, a(R_0)

   ST b, R_1
10. What is a DAG?

Part-B (5 × 10 = 50 Marks)

11. a) Explain the various phases of a compiler with a neat diagram. [6]
    b) Give a brief description about LEX. [4]

12. a) Write the rules to compute FIRST and FOLLOW. [4]
    b) Construct Predictive parser for the following grammar. Justify your design with an example. [6]
       E → E + T | T
       T → F * F | F
       F → (E) | id

13. a) Construct a DAG for the following expression: a+a*(b-c) + (b-c)*d [4]
    b) Translate the following expression into quadruple, triple and indirect triple representations. [6]
       a[i] = b*c − b*d

14. a) Write short note on Code optimization from DAGs. [4]
    b) Discuss about semantics-preserving transformations with examples. [6]

15. a) Describe the Issues in the design of a Code Generator. [4]
    b) Discuss about peephole optimization. [6]

16. a) Describe the role of Lexical Analyzer. [4]
    b) Construct SLR parsing table for the following grammar. [6]
       E → E + T | T
       T → F * F | F
       F → (E) | id

17. Write short notes on any *two* of the following:
    a) Rules for Type Checking.
    b) Stack allocation strategies. [5]
    c) Simple code generator. [5]
    [5]

ఔఴఴఴఴఴఴ