**Q-Bank for II CIA Portion**

**part A**

1. Given the following derivation, construct the corresponding parse tree : S⇒ AB⇒ BeB ⇒ deB⇒deaBc⇒deadc
2. What is meant by type checking?
3. Write the uses of ad-hoc syntax directed translation.
4. What is meant by an abstract syntax tree?
5. What is meant by scope or namespace?
6. How will you eliminate redundancies using value numbering?
7. Give two examples for inter procedural optimization.
8. Distinguish between Inherited and synthesized attributes.
9. Distinguish between ad-hoc syntax directed translation and attribute grammar.
10. Write some important properties of Intermediate Representation.
11. What is meant by an extended basic block?
12. Why do we need global methods in code optimization?
13. When will you say that a dominates b in a Control Flow Graph?
14. Write the various steps for building SSA form.
15. Define CFG and its use.
16. Explain the use of type systems.
17. Define synthesized attribute.
18. Define Circular Attribute grammar.
19. Explain benefits of DAG.
20. Write the attribute grammar for the Signed Binary Numbers (SBN).
21. List out the components of Type systems.
22. What is inherited attribute.
23. Differentiate parse tree with Abstract Syntax Tree. Give example.
24. Represent the following expressions in stack machine code : x-2\*y.
25. Define SSA.
26. Explain Virtual registers.
27. Draw the array layout for the array A[1..2,1..4] as row major order.
28. Write down the rules to evaluate the assignment operator.
29. Write the equivalent ILOC code for the string assignment statement a[1]=b[1].
30. Represent the following examples in the stack machine code: x-2\*y
31. Explain the benefits of DAG
32. What is the difference between attribute grammar and syntax directed translation?
33. Write any two advantages of three address code statements.
34. List the interface routine and how it is used by the symbol table.
35. What are the various ways to pass a parameter in a function?
36. What are the methods of translating Boolean expressions?
37. What are the properties of Optimizing compiler?
38. What is Local value numbering?
39. What are the various methods for allocating activation records?
40. Define dead-code elimination with an example.
41. Define parse tree. Give an example.
42. What are the two ways to perform context sensitive analysis?
43. List out the problems with attribute grammar approach.
44. What is meant by an abstract syntax tree?
45. What is control flow graph?
46. Give the approaches used to manage namespace of fieldnames in symbol table. Which method is more advantageous? Why?
47. Define Control abstraction.
48. How are values communicated between procedures?
49. Give the 3 schemes of array storage layout.
50. What are the issues to be considered in choosing between branching and predication in the implementation of if-then-else?
51. What are the issues to be considered for optimization?
52. Define local methods of optimization?
53. Define live variables.
54. List out the interface routine and its use in symbol table.
55. What are the different flavours of SSA form? In what way do they differ?
56. Consider the grammar

S->(L)|a

L->L,S|S

Find the parse tree for the sentence (a, a)

1. Define handle.
2. Give the principles followed in ad-hoc approach to build abstract syntax tree.
3. What are the various categories of intermediate representation?
4. Compare the two memory models of compiler.
5. Define Namespace.
6. How do the memory models used have impact on code shape?
7. What is short-circuit evaluation?
8. State the goal of code optimization.
9. Compare debugging compiler with optimizing compiler.
10. What is meant by inter-procedural constant propagation and inline substitution?
11. How is live-variable used by a compiler?
12. What is call graph?

**part – B**

**Theory Questions**

1. Explain ad-hoc syntax directed translation with an example.
2. Explain Global Redundancy Elimination with an example.
3. What is meant by an attributed grammar? Explain the various methods for performing evaluation. Explain with an example.
4. Explain the Linear Intermediate Representation with an example.
5. Explain a method for building static single assignment form.
6. Generate three address code for the following expressions and represent in quadruple,triple and indirect triple.  
    (a) if (a<b) z=30; else z=40; x=90;

(b) while(I>0){v=v\*I;I=I-1;}

1. Explain in detail about the Static Single Assignment form.
2. Generate the three address code for the following expressions and represent in a quadruple, triple and indirect triple:

v1=50, v2=30, v3=v1+v2, v4=v1+34-(45+v3+v2)

1. a) Compare Adhoc Syntax directed translation with attribute grammar.
2. Explain adhoc syntax directed translation method for signed binary numbers (SBN)
3. Explain in detail about Global redundancy elimination.
4. a)Write the grammar for unsigned binary numbers and write the attribution rules to compute the integer value for a binary number.

b) Write the SDT scheme for the expression grammar to specify the translation of an expression into postfix notation.

1. Why is a symbol table used by a compiler? Explain the routines to build a symbol table and to handle nested scopes and various approaches for managing the namespace of field names.
2. Explain the concept of removing redundant expressions in:

Super-local value numbering

1. Explain in detail about static single assignment form.
2. Explain ad-hoc syntax directed translation with an example.
3. Explain the control flow graph and dependence graph with suitable example.
4. Explain iterative data-flow analysis, its limitations and data flow problems.
5. Explain static single assignment form.
6. Explain in detail about symbol table and its construction.
7. Loop Unrolling
8. Inline substitution

Numerical Problem Questions

1. Ad-hoc SDT
2. IR( 3AC, 2AC, Quadruple, Triple, Indirect triple, CFG, Dependence Graph, AST, DAG, Parse tree)
3. SSA
4. Linear IR to CFG(leader selection)
5. Local Value Numbering
6. **Tree height balancing**
7. Superlocal value numbering
8. Loop Unrolling
9. Global Optimization(LiveOut)
10. **Global Code placement**
11. Procedure placement