Code Optimization

1. With an example each explain the following loop optimization techniques:
   (i) Code motion (ii) Induction variable elimination and (iii) strength reduction (10)
2. Explain the optimization of basic blocks. (5)
3. Explain different code optimization techniques available in local and global optimizations? (10)
4. How the optimization of basic blocks is done by a compiler? (6)
5. Write the algorithm for partitioning a sequence of three-address instructions into basic blocks. (4)
6. Write the algorithm for identifying the basic blocks from a sequence of three address code statements. (5)
7. Construct the DAG for the following basic block (5)

   D := B * C
   E := A + B
   B := B * C
   A := E – D

8. How do algebraic laws help in optimizing basic blocks? (4)
9. With an example, explain copy propagation and dead code elimination. (4)
10. Explain the principal sources of optimization (10)
11. Explain optimization of basic blocks (5)
12. With suitable examples explain loop optimization. (5)
13. Explain copy propagation with an example. (4)
14. Explain common sub expression elimination with an example. (4)
15. With suitable example of a basic block, explain the code-improving transformations of a basic block. (6)
16. Explain code motion with an example (3)
17. Write the algorithm for partitioning a sequence of three-address instructions into basic blocks (3)
18. With suitable examples explain loop optimization techniques. (7)
19. With suitable examples of a Basic Block, explain code improving transformations of a basic block. (7)
20. With suitable examples explain the following loop optimization techniques:
    (i) Code motion (ii) Induction variable elimination and (iii) strength reduction (7)
21. Explain the optimization of basic blocks. (7)

Code Generation

1. Explain any two issues in the design of a code generator. (5)
2. Write the Code Generation Algorithm and explain the *getreg* function. (6)
3. Generate a code sequence for the assignment d=(a-b)+(a-c)+(a-c) (4)
4. Explain the code generation algorithm. Illustrate with an example. (6)
5. State the issues in design of a code generator. (4)
6. Write the code generation algorithm. (6)
7. For the following C statement, write the three-address code.
   X := A – B + C – D + E – F
   Convert the three-address code into machine code. (10)
8. Write the code generation algorithm. (5)
9. Explain the problems that might be encountered during code generation. (6)
10. Translate the expression W := (A-B) + (A-C) + (A-C) into three address code sequence and then generate the machine code for the three address code. (10)

11. How do you derive the cost of an instruction? Illustrate with a sample code fragment. (5)
12. Explain issues in design of a code generator (5)
13. Explain simple code generation algorithm (5)
14. Identify any four issues in the design of a Code Generator. (6)
15. Write the code generation algorithm. Using this algorithm generate code sequence for the expression x = (a – b) + (a + c). (10)
16. Write the code generation algorithm. Using this algorithm, generate the code sequence for the expression x = (a – b) + (a + c)+(a +c)
17. Explain any three issues in the design of a code generator. (3)
18. For the following C statement, write the three-address code and quadruples.
S:= A – B + C * D – E + F
Also convert the three-address code into machine code. (8)
19. Write the Code Generation Algorithm and explain the *getreg* function. (6)