

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Question Paper Code : 70744**

M.C.A. DEGREE EXAMINATION, MAY/JUNE 2013.

Elective

MC 9293/MC 993 – COMPILER DESIGN

(Regulation 2009)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. List some compiler construction tools.
2. List down the functions performed in synthesis phase.
3. Give the algebraic properties of regular expressions.
4. Left factor the grammar  $S \rightarrow Aa|Ab|Ac$ .
5. Define augmented grammar.
6. Using numerical representation method translate the following Boolean expression  $A < B$  or  $C$ .
7. How are leaders identified for basic blocks?
8. What is constant folding, Give a suitable example.
9. List some applications of DAG?
10. Differentiate static allocation from stack allocation.

PART B — (5 × 16 = 80 marks)

11. (a) Explain the various phases of compiler with suitable examples. Also write about the errors handled in each phase. (16)

Or

- (b) Draw NFA for the regular expression  $(0|1)^*1(0|1)^*$ . Obtain the equivalent DFA and minimize its number of states. (16)



rejinpaul.com

Grow With Us



12. (a) Consider the following grammar for list structures.

$$S \rightarrow a^{\wedge}(T)$$

$$T \rightarrow T, S | S$$

- (i) Find right most derivation for  $(a, (a, a))$  (4)
- (ii) Indicate the handle of each right sentential form. (4)
- (iii) Show the steps of shift reduce parser corresponding to these right most derivations. (4)
- (iv) Construct the parse tree for the above. (4)

Or

- (b) (i) Construct Predictive parsing table for the grammar (10)

$$S \rightarrow aAd$$

$$A \rightarrow BC$$

$$B \rightarrow b|\epsilon$$

$$C \rightarrow c|\epsilon$$

And parse for the string  $w = abcd$ .

- (ii) Differentiate regular expressions from context free grammars. (6)
13. (a) (i) Write the translation scheme for producing three address codes for assignment statements with integer data types. (8)
- (ii) Using the above translation schemes and generate three address code for  $A := (A - B) * (C + D)$ . (8)

Or

- (b) (i) Explain Backpatching in detail. (10)

- (ii) Translate the following assignment statement into

(1) Quadruples

(2) Triples

(3) Indirect Triples  $p = q * -r + q * -r$ . (6)

14. (a) Consider the following program block

$I := 1$

While  $(I \leq 10)$  begin  $c[I] := 0; I := I + 1$ ; end;

$I := 1$

do begin

if  $(c[I] := 0)$  then  $c[I] := 0$ ;





$c[I] := (A[I] + B[I]) * (A[I] + B[I]);$

$I := I + 1;$

end

while ( $I \leq 10$ );

- (i) Generate three address code after removing the dead code if available and merging loops if possible. (4)
- (ii) Identify the basic blocks and construct flow graph. (4)
- (iii) Identify the loop invariant computations and move them to the pre header of the loop. (4)
- (iv) Eliminate the induction variables, common sub expressions and find how much optimization is obtained. (4)

Or

- (b) (i) Explain various loop optimization techniques with examples. (8)
  - (ii) Write short notes on principal sources of optimization. (8)
15. (a) (i) Write short notes on issues in the design of code generator. (8)
- (ii) Construct DAG for  $a + a * (b - c) + (b - c) * d$ . (8)

Or

- (b) Formulate a simple code generation algorithm. Show how the algorithm works for the following code (16)

$x = a - b$

$y = a - c$

$z = x + y$

$d = x + z$ .

