



QP CODE: 26802597



26802597

Reg No :

Name :

IMCA DEGREE EXAMINATION JUNE 2026

Eight Semester

INTEGRATED MCA

CORE - IMCA8C04 - COMPILER DESIGN

2020 Admission Onwards

CE6AE78D

Time: 3 Hours

Maximum: 75 Marks

Part A

*Answer any **ten** questions*

Each question carries 3 marks

1. Define interpreters. Give its functions.
2. Explain how symbol tables are used in a compiler.
3. Describe the scanner generators and parser generators.
4. Create ϵ -NFA for the regular expression a^* .
5. Summarize about LEX tool.
6. What are the advantages and disadvantages of operator precedence parser?
7. *Eliminate the left recursion from the grammar.*
 $E \rightarrow E+T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$
8. Describe the process of constructing the canonical collection of LR(0) items.
9. How does SLR(1) parsing differ from LR(0) parsing?
10. Explain the concept of Directed Acyclic Graph (DAG) representation of basic blocks. What advantages does it provide?
11. What is meant by optimizing basic blocks? Provide an example of a basic block optimization.
12. What are common subexpression elimination and its role in code optimization?

(10×3=30 marks)





Part B

Answer **all** questions

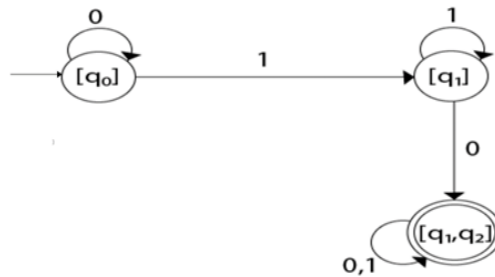
Each question carries **9** marks

13. a) Outline the front end and back end of a compiler.

OR

- b) Describe the steps involved in the recognition of tokens by a lexical analyzer.

14. a) Convert the following NFA into its equivalent DFA.



OR

- b) Discuss the role of parse trees in understanding ambiguous grammars. How can multiple parse trees for the same input string illustrate the ambiguity present in a grammar?

15. a) Describe about various conflicts in shift reduce parser.

OR

- b) Explain predictive parsing algorithm in detail.

16. a) What are the different types of errors that can occur in the compilation process? Discuss strategies for error detection and reporting during both syntax analysis and semantic analysis.

OR

- b) Describe the process of using a canonical LR parsing table to parse an input string. Illustrate your answer with an example that shows state transitions, shift actions, and reduce actions.

17. a) Discuss the Directed Acyclic Graph (DAG) representation of basic blocks. How does the use of DAGs enhance the optimization capabilities of a compiler? Provide examples of how DAGs can simplify code generation and optimization.

OR

- b) Describe simple code generation methods used in compilers. Compare and contrast these methods with more complex code generation strategies, highlighting their advantages and limitations.

(5×9=45 marks)

