(c) Write a simple code generator algorithm. With an example code, show how the algorithm generates code.

(d) Construct SLR parsing table for the following grammar :

$$E \rightarrow E + T \mid T$$
$$T \rightarrow TF \mid F$$
$$F \rightarrow F^* \mid a \mid b$$

★★★

2023

BACHELOR OF COMPUTER APPLICATION

Paper : BCA–HE–6036

(Systems Programming)

Full Marks : 60
Pass Marks : 24

Time : 3 hours

The figures in the margin indicate full marks for the questions

1. Choose the correct option (any *five*) : 1×5=5

(a) How many components are there in a context-free grammar?

(i) 2

(ii) 3

(iii) 4

(iv) 5

(b) A set of tokens is known as

(i) non-terminals

(ii) start symbol

(iii) terminal symbols

(iv) productions

(c) Which of the following components is important for semantic analysis?

(i) Yacc

(ii) Type checking

(iii) Lex

(iv) Symbol table

(d) Which phase of the compiler is also known as scanner?

(i) Syntax analysis

(ii) Semantic analysis

(iii) Lexical analysis

(iv) Code generation

(e) Which of the following errors can a compiler check?

(i) Logical error

(ii) Syntax error

(iii) Both logical and syntax errors

(iv) Compiler cannot check error

(f) Which phase of the compiler is known as parser?

(i) Syntax analysis

(ii) Semantic analysis

(iii) Lexical analysis

(iv) Code generation

(g) A lexical analyzer's output is

(i) intermediate code

(ii) a parse tree

(iii) a stream of tokens

(iv) machine code

(h) Which computer program accepts the high-level language and converts it into assembly language?

(i) Interpreter

(ii) Linker

(iii) Assembler

(iv) Compiler

(i) Which grammar gives multiple parse trees for the same string?

(i) Unambiguous

(ii) Regular

(iii) Ambiguous

(iv) Context-free

(j) Keywords are recognized in a compiler during

(i) lexical analysis

(ii) syntax analysis

(iii) semantic analysis

(iv) code generation

2. Answer the following questions (any *five*) : 2×5=10

(a) What is Yacc?

(b) What are the three storage allocation strategies?

(c) What are the various parts in LEX program?

(d) What is symbol table?

(e) What is an interpreter?

(f) What is absolute loader?

(g) Define relocatable program.

3. Answer the following questions (any *five*) : 5×5=25

(a) Explain relocating loader and dynamic linking.

(b) What is a syntax tree? Describe the construction of syntax trees for expressions.

(c) Differentiate among tokens, patterns and lexeme.

(d) Differentiate between loader and linker.

(e) Differentiate between top-down and bottom-up parsing.

(f) Write the disadvantages of lexical analysis.

(g) Explain three-address code generation.

(h) Explain stack allocation for runtime storage management.

(i) Explain activation records in the context of storage organization.

4. Answer the following questions (any *two*) : 10×2=20

(a) Explain the phases of compiler with a neat diagram.

(b) Construct a syntax directed translation schema that translates arithmetic expression from infix to postfix notation.