USN ☐☐☐☐☐☐☐☐☐☐ **18CS61**

Sixth Semester B.E. Degree Examination, June/July 2024
System Software and Compilers

Time: 3 hrs.   Max. Marks: 100

Note: *Answer any FIVE full questions, choosing ONE full question from each module.*

Module-1

1 a. Explain architecture of SIC/XE machine. **(10 Marks)**

   b. Generate Target address for the following object codes :
      i) 032600
      ii) 03C300
      Contents of X = 000090   B = 006000   PC = 003000 **(04 Marks)**

   c. Define control section. Explain EXTDEF and EXTREF with examples. **(06 Marks)**

OR

2 a. Write steps and algorithm of pass 2 of two pass assembler. **(10 Marks)**

   b. Give the general formats of the following records :
      i)   Header Record
      ii) Text Record
      iii) End Record **(06 Marks)**

   c. Write the algorithm for absolute loader. **(04 Marks)**

Module-2

3 a. With the help of diagram, explain the various phases of a compiler. **(08 Marks)**

   b. Write the transition diagram to recognize the token below.
      i) Identifier
      ii) Relational operator. **(06 Marks)**

   c. Explain the concept of input buffering in the lexical analysis. **(06 Marks)**

OR

4 a. Explain interaction between lexical analyzer and parser with the help of diagram. **(08 Marks)**

   b. What are the applications of compiler technologies? Explain any two. **(06 Marks)**

   c. What are operations on Languages? List and explain. **(06 Marks)**

Module-3

5 a. Write an algorithm to eliminate left recursion. Elimination left recursion from grammar.
      S → Aa | b
      A → Ac | sd | E **(08 Marks)**

   b. Give rules for constructing FIRST and FOLLOW sets. **(06 Marks)**

   c. List actions of shift Reduce parser. Show the actions for input string w = id * id using the grammar.
      E → E + T | T
      T → T * F | F
      F → (E) | id **(06 Marks)**

OR

6 a. Explain ambiguity in "dangling else" grammar. How do you eliminate it? Explain. **(10 Marks)**

b. Construct predictive parting table by making necessary changes to the grammar given below and parsing string w = id + id **(10 Marks)**

$E \rightarrow E * T | T$

$T \rightarrow id + T | id$

Module-4

7 a. With an example program, explain the structure of a LEX program. **(06 Marks)**

b. What is regular expression? Explain any 8 characters that form a regular expression. **(10 Marks)**

c. Explain the use of yywrap( ) function. **(04 Marks)**

OR

8 a. Explain shift Reduce parser with an example. **(10 Marks)**

b. Write a YACC program to evaluate arithmetic expression involving operators +, −, *, /. **(10 Marks)**

Module-5

9 a. Write SDD for simple disk calculator, and give annotated parse tree for 3 * 5 + 4n. **(10 Marks)**

b. Construct syntax tree and DAG for the expression
$a + a * (b − c) + (b − c) *d$ **(06 Marks)**

c. Define synthesized and inherited attributes with example. **(04 Marks)**

OR

10 a. Explain design issues in code generation. **(08 Marks)**

b. Explain the following with examples :
  i) Quadruples
  ii) Triples
  iii) Indirect Triples **(09 Marks)**

c. Write the machine instruction for the following three address instruction x = y − z. **(03 Marks)**

* * * * *