CBCS SCHEME

USN □□□□□□□□□□ **18CS61**

Sixth Semester B.E. Degree Examination, Feb./Mar. 2022
System Software and Compilers

Time: 3 hrs. Max. Marks: 100

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

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

VTU EXAMS FEB./MAR. 2022

Module-1

1 a. Explain the following with concept to SIC/XE machine architecture :
i) Registers ii) Data formats iii) Instruction formats iv) Addressing modes. **(10 Marks)**

b. What is a loader? What are the basic functions the loader has to perform? **(03 Marks)**

c. What is program relocation? Illustrate how modification record is used in relocation of the program. **(07 Marks)**

OR

2 a. Generate the object code for the following SIC/XE source program.

```
SUM     START   0
FIRST   CLEAR   X
        LDA     #0
        +LDB    #TOTAL
        BASE    TOTAL
LOOP    ADD     TABLE, X
        TIX     COUNT
        JLT     LOOP
        +STA    TOTAL
COUNT   RESW    1
TABLE   RESW    2000
TOTAL   RESW    1
        END     FIRST
```

CLEAR = B4, LDA = 00, LDB = 68, ADD = 18, TIX = 2C, JLT = 38, STA = 0C **(10 Marks)**

b. State SIC assembler directives. **(03 Marks)**

c. Explain in detail program blocks. **(07 Marks)**

Module-2

3 a. Show a typical language processing system with a diagram. **(06 Marks)**

b. Write an algorithm for "look ahead code with sentinels" **(04 Marks)**

c. Relate handling reserved words and identifiers during recognition of tokens. Illustrate the transition diagrams for the following: i) Relop ii) unsigned number. **(10 Marks)**

OR

4 a. State the three types of software productivity tools. **(06 Marks)**

b. Explain the role of lexical analyzer. **(04 Marks)**

c. Explain regular expression, extended regular expression and regular definition. Write the regular definition of id and int. **(10 Marks)**

Module-3

5 a. Eliminate left recursion from the following grammar S → A ; A → aB|ac|Aab|Ade ; B → bBc|ef. **(04 Marks)**

b. Explain how the error recovery strategies in predictive parsing can be realized with example. **(08 Marks)**

c. For the grammar: S → (L) |a L → L, S|S. Construct predictive parsing table and show the moves made by the parser on the input (a, (a, a)). **(08 Marks)**

OR

6 a. Write an algorithm for a recursive descent parser. **(04 Marks)**

b. What is ambiguous grammar? Prove the following grammar is ambiguous :
E → E + E| E * E| (E)|id and write its equivalent un-ambiguous grammar with proper reasoning. **(08 Marks)**

c. Explain the possible four actions of SR parser. For the following grammar S → 0S1|01, show bottom up parsing for the given string "000111". **(08 Marks)**

Module-4

7 a. Show the structure of a lex program. **(04 Marks)**

b. State any three LEX – YACC variables and functions. **(06 Marks)**

c. Demonstrate the following regular expressions with example :
i) [ ] ii) { } iii) / iv) ( ) v) ^ **(10 Marks)**

OR

8 a. What are LEX and YACC? **(04 Marks)**

b. Explain regular expressions to identify the following :
i) identifier ii) negative integer iii) positive number. **(06 Marks)**

c. Write a YACC program to design a simple calculator. **(10 Marks)**

Module-5

9 a. Differentiate synthesized and inherited attributes. Obtain SDD for simple declaration statement. Construct a dependency graph for "float a, b, c" and give its evaluation order. **(08 Marks)**

b. Obtain the directed acyclic graph for the expression x + x *(y + z) + (y + z) * w, along with the steps. **(06 Marks)**

c. Explain the common three – address instruction forms. **(06 Marks)**

OR

10 a. Define SDD and its types. Write SDD for simple desk calculator and give annotated parse tree for the expression (7 – 2)*(8 – 1)n. **(08 Marks)**

b. Translate the assignment statements: a = b[i] + c[j] into :
i) a syntax tree ii) quadruples iii) tripples.

c. Generate code for the following three – address statements assuming a and b are arrays whose elements are 4-byte values.

x = a[i]
y = b[j]
a[i] = y
b[j] = x **(06 Marks)**

* * * * *

VTU EXAMS FEB / MAR 2022