CBCS SCHEME

USN | | | | | | | | | | |

17CS63

Sixth Semester B.E. Degree Examination, July/August 2022
System Software and Compiler Design

Time: 3 hrs. Max. Marks: 100

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

Module-1

1 a. Explain Memory, Registers , Data formats , Instruction formats and Addressing modes of SIC/XE machine. **(08 Marks)**

b. Generate the complete object program for the following SIC/XE assembly code. Given LDA = 00 , CLEAR = B4 , LDB = 68 , ADD = 18 , MUL = 20 , LDS = 6C , DIVR = 9C , JEQ = 30.

```
SSCD    START   5000
FIRST   LDA     # 11
        CLEAR   X
        LDB     @ TWO
        BASE    TWO
        ADD     K100
        + MUL   M300
LOOP    LDS     TWO, X
        DIVR    S, A
        JEQ     LOOP
TWO     WORD    2
M300    RESW    300
K100    BYTE    X '00'
        END     FIRST
```

(12 Marks)

OR

2 a. Write an algorithm of PASS – 1 of two pass assembler. **(08 Marks)**

b. Calculate the Target Address for the following machine instructions. Given (X) = 000690 (B) = 006030 (PC) = 003060
i) 032600 ii) 026030 iii) 0310C303 iv) 010030. **(04 Marks)**

c. Write an algorithm of a One – Pass Macro processor. **(08 Marks)**

Module-2

3 a. By recalling the concepts of loading the relocatable program into main memory. Write an algorithm for Absolute loader. **(04 Marks)**

b. Explain a Bootstrap loader for SIC/XE machine in detail. **(08 Marks)**

c. Explain Dynamic linking with suitable diagrams. **(08 Marks)**

OR

4 a. Apply the knowledge of program linking and write an algorithm of Pass 1 and Pass 2 of a linking loader. **(12 Marks)**

b. Explain the features of MS – DOS linker. **(08 Marks)**

Module-3

5 a. Apply the knowledge of compilers to illustrate the various stages of compilation for a given source code : MUL = a * b + 5 in detail. **(10 Marks)**

b. Explain the Input buffering technique used by Lexical analyzer. **(06 Marks)**

c. Explain the structure of Lex program. **(04 Marks)**

OR

6 a. Sketch the transition diagram and pseudocode for RELOP (< , <= , > , >= , <> , =). **(10 Marks)**

b. Formulate the Algebraic laws of Regular Expression. **(05 Marks)**

c. Explain Software productivity tools. **(05 Marks)**

Module-4

7 a. Define Ambiguity. Show that the following grammar is ambiguous for the input : a / b * c
R → R ' I ' R | RR | R* | (R) | a | b | c

i) Give an unambiguous grammar given the precedence of operators from lowest to highest are concatenation * I ( ) identifier.

ii) Construct the parse tree using an unambiguous grammar for the above given input string. **(06 Marks)**

b. Eliminate left recursion for the following grammar
S → S + T | S T | (S) | S * | a
T → (U) | a
U → U, T | T. **(04 Marks)**

c. Construct the predictive parsing table for the given grammar and parse the input : abcd.
S → aAcd | B Ce A → b | ∈
B → C f | d C → f e | C **(10 Marks)**

OR

8 a. Develop an algorithm to construct the table driven predictive parsers with a neat diagram. **(06 Marks)**

b. Build the shift reduce parser for the given grammar on input : abbcdc
S → a A c B e A → Ab | b B → d. **(06 Marks)**

c. Construct the operator precedence relation table for the given grammar
E → EAE | (E) | id , A → * | + and parse the input (id + id * id). **(08 Marks)**

Module-5

9 a. Build the syntax Directed Definition for a Simple Desk calculator and apply this to construct an Annotated Parse tree and Dependency graph for 3 * 5 + 7n. **(10 Marks)**

b. Explain the design issues of a code generator. **(10 Marks)**

OR

10 a. Construct the Directed Acyclic Graph and identify the value numbers for the sub expressions of the following expressions , assuming + associates from the left
i) a + b + (a + b) ii) a + b + a + b. **(06 Marks)**

b. Translate the arithmetic expression a + - (b + c) into i) Syntax tree ii) Quadruples iii) Triples iv) Indirect Triples. **(08 Marks)**

c. Generate machine instruction for the following three address statements :
i) y = b * c ii) z = * c iii) c = b + a[i] iv) if a < b goto L. **(06 Marks)**

* * * * *