BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(MID SEMESTER EXAMINATION)

CLASS: BE   SEMESTER: VI
BRANCH: CSE   SESSION : SP/2019

SUBJECT : CS6105 COMPILER DESIGN

TIME: 1.5 HOURS   FULL MARKS: 25

INSTRUCTIONS:
1. The total marks of the questions are 30.
2. Candidates may attempt for all 30 marks.
3. In those cases where the marks obtained exceed 25 marks, the excess will be ignored.
4. Before attempting the question paper, be sure that you have got the correct question paper.
5. The missing data, if any, may be assumed suitably.

Q1 (a) Write tasks performed by two cousins of Compilers. Explain with example how they   [2]
        differ from simple compiler?
    (b) Explain the various phases of a compiler for the given program: [3]

        Program
        var I,j: integer;
        var r :real;
        begin
        r=i+j*10;
        end.

Q2 (a) Discuss the challenges in compiler design and its applications.   [2]
    (b) Write a LEX specification file to identify the tokens of the language C.   [3]

Q3 (a) Which of the following expressions have l-values and / or R-values.   [2]
        (i) A[I+1]   (ii) * A   (iii) & A   (iv) &(* A)   (v) *(& A)
        (vi) * (&(&A))

    (b) Consider the following grammar: [3]
        Stmt -> if cond then stmt else stmt
              | if cond then stmt
        Show this grammar is ambiguous
        Construct an equivalent unambiguous grammar

Q4 (a) Write an algorithm for eliminating left recursion. Verify your algorithm with the   [2]
        following grammar
        S->Aa
        A->Sb|c
    (b) Write an algorithm for Recursive Descent Calculator of the following grammar   [3]
        E->E+T|T
        T->T*F|F
        F->(E)|id

Q5 (a) Show that the given grammar is not in LL(1)   [2]
        S→ A|B,   A→cA+b|a,   B→cB+a|b
    (b) Consider the grammar   [3]
        E→ BA
        A→&BA|€
        B→true|false
        Show that the grammar is LL(1) and construct the predictive parsing table.

Q6   Consider the following grammar: [5]
Non-terminals {S, A}
Terminals {c, b}
Production rules:
{   S→ AA,
        A→cA,
        A→b   }
Starting symbol is 'S'.
Construct SLR parsing table for this grammar. List out all conflicts which are present in the designed SLR parsing table?
For removal of above conflicts (if they are present), what will be done? Explain in detail.