United College of Engineering and Research
First Sessional Examination
Compiler Design(RCS-602)
B.Tech 6th Sem(CS/IT)

Time 2hrs. MM.30

Note : Attempt All Section

Section A [Attempt all parts] **[1X10=10]**

1. What is cross compiler?
2. Differentiate between Compiler and Interpreter?
3. A C++ translator written in Java-language that takes C++ code and produce Java as output. Create a C++ translator in C for same.
4. Draw the transition diagram for relational operator?
5. Find the no of Lexeme in given code fragment

```
int main()
{
        int a=10,b=20;
        printf("sum is %d",a+b);
        return 0;
}
```

6. The regular expression 0*(10*)* denotes the same set as
   a) 0 + (0 + 10)* b). (0 + 1)*10(0 + 1)* c) (1*0)*1* d). None of these
7. Differentiate between linker and loader?
8. State any two reasons as to why phases of compiler should be grouped.
9. Explain the term token, pattern and lexeme.
10. Write the structure of lex programming. Give the example.

Section B [Attempt any 5 parts] **[2X5=10]**

1. Discuss input buffering and preliminary scanning in lexical analysis
2. Explain recursive decent parser. Create the parser for the following the following grammar
   E→ iE'
   E' → +iE' |€
3. Eliminate left recursion from the following grammar
   S→AB
   A→BS | b
   B →SA | a
4. Design a deterministic finite automata(DFA) for accepting the language
   L= { $a^n b^m$ | n + m= even}
5. What is ambiguous grammar? Explain with example. Write the rule to convert ambiguous grammar into unambiguous grammar.
6. What is Non deterministic grammar? Is this grammar is suitable for L L(1) parsing or not, and also define how to remove Non-determinism from the grammar.

Section C [Attempt any 2 parts] **[5X2=10]**

1. Explain non recursive predictive parsing. Consider the following grammar construct predictive parsing table
   E→TE'
   E'→+TE'
   T→FT'
   T'→*FT'|€
   F→F^*|a|b

2. Explain the phases of the compiler in detail. Write down the output of each phase for the expression a=b*c + 50.
3. Design optimize DFA for the R.E. (0|1)* 0(011)* by applying Thomson construction rule?