

**COMPILER DESIGN**

(Computer Science and Engineering)

Time: 3 hours

Max Marks: 70

Answer any FIVE questions  
All questions carry equal marks

\*\*\*\*\*

- 1 (a) Explain in detail the role of lexical analyzer with possible error recovery actions.  
(b) What is compiler? Explain various phases of compiler in detail with neat sketch.
- 2 (a) Eliminate ambiguity if any, from the following grammar for Boolean expressions.  
 $prbex \rightarrow bexpr \text{ or } bterm|bterm$   
 $bterm \rightarrow bterm \text{ and } bfactor|bfactor$   
 $bfactor \rightarrow nst \text{ factor} |(bexpr)|true|false.$   
 Where or, and, not ( , ), true, false are terminals in the grammar.  
 (b) Write a recursion descent parser for the above grammar.
- 3 (a) What are the advantages of LALR parser technique?  
(b) Distinguish SLR and LALR grammar.
- 4 (a) List out some typical semantic errors. Explain how they can be rectified.  
(b) What is static checking? Give some examples of static checks.
- 5 (a) Explain in detail about the contents of symbol table.  
(b) Distinguish between static scope and dynamic scope. Briefly explain access to non-local names in static scope.
- 6 (a) Explain with example local optimization technique.  
(b) Why code optimization phase is optional in compiler? Justify your answer.
- 7 Write notes on the following:  
(a) Induction variable.  
(b) Global data flow analysis.

- 8 Generate target machine code for the following program.

```

ain() m
{
    for( j=2; j<=n; j++)
        a[j]=1;
    count=0;
    for(j=2; j<=n**0.5;j++)
        if(a[j])
        {
            count++;
            for(k=2*j; j<=n; k=k+j)
                a[k]=0;
        }
    printf("%d",count);
}

```

\*\*\*\*\*