

Time: 3 Hours

Total Marks: 80

- N.B: (1) Question No. 1 is compulsory.
 (2) Attempt any three questions out of the remaining five questions.
 (3) Figures to the right indicate full marks.
 (4) Make suitable assumptions wherever necessary.

- Q.1. A. Explain the two-layered model of system software. Justify the need for machine-dependent and machine-independent system programs. 5
 B. Construct operator precedence Parser for the grammar: 5
 $E \rightarrow E+E \mid E * E \mid a$.
 Parse the string "a+a*a" using the same parser.
 C. Define Relocation and Linking. Why are they crucial functions of a linking loader 5
 D. Define Ambiguous Grammar. How is ambiguity a problem in the design of a parser? 5
- Q.2. A. With reference to the LR parsing family, differentiate between SLR (1), CLR (1), and LALR (1) parsers based on the size of the parsing table and the power of the grammar they can handle. 10
 B. Construct Three address code for the following program 10

```

i = 1;
x = 0;
while (i <= n)
{
    x = x + 1;
    i = i + 1;
}
    
```
- Q.3. A. Explain the various data structures used in the Two-Pass Assembler design, specifically focusing on the Opcode Table, Symbol Table, and Literal Table. 10
 B. Explain the various features of a Macro facility, such as nested macro calls and conditional macro expansion, with suitable examples. 10
- Q.4. A. Explain the working of a Single-pass macro processor with neat flowchart. 10
 B. What are the different ways of representing Intermediate code? Explain with suitable example. 10
- Q.5. A. Explain any five code optimization techniques in compiler designing with suitable example 10
 B. Explain design issues in code generation in detail. 10
- Q.6. A. Differentiate between Machine-Dependent and Machine-Independent code optimization techniques. Give two distinct examples for each category. 10
 B. Explain advanced macro facilities with suitable examples. 10
