**UNIT I**

1. Define system software.
2. List types of software system.
3. Define Loader, Assembler, Macroprocessor, Compiler, Interpreter.
4. List down assembler directives.
5. What is syntax and function of EQU and ORIGIN statement. What is the limitation/constraint of these statements.
6. What is the function of LTORG statement.
7. What is the significance of LTORG statement.
8. What is the difference between constant and literal?
9. How does single pass assembler work? What is Table of Incomplete Instructions(TOII)?
10. What are the functions of Pass I of two pass assembler.
11. What are the functions of Pass II of two pass assembler.
12. Which are the phases of language processor.
13. All symbols in a ALP are forward referenced. True or False?
14. All literals in a ALP are forward referenced. True or False?
15. Is a macro open or closed sub routine.
16. Does macroprocessing happen before or after compilation / assembling of a program?
17. List down all macro directives. Do macro directives occur in expanded code?
18. What are the functions of Pass I of two pass macroprocessor.
19. What are the functions of Pass II of two pass macroprocessor.
20. Can II pass macroprocessor handle nested macros?

**UNIT II**

1. What is a loader?
2. What is compile and go loader.
3. List down functions of loader.
4. List down types of loader.
5. Which functions are done by whom in
6. Absolute Loader
7. Relocating Loader
8. Direct Linking Loader.
9. What is the input to Pass I of two pass DLL? What is output?
10. What is the input to Pass II of two pass DLL? What is output?
11. What is the information provided to loader by assembler in a relocating loader?
12. What is the information provided to loader by assembler in a DLL?
13. Which is the best type of loader?

**UNIT III**

1. List down phases of compiler.
2. What is the function of each phase?
3. What is lex?
4. How are tokens identified in lex?
5. List down the types of tokens?
6. What is the structure of lex file?
7. RE to DFA algorithm is implemented in which phase of compiler?
8. How do lexical analyzer and parser work together?
9. What is uniform symbol table?
10. Why follow positions in RE to D
11. FA are found only for “.” and “\*” operator?

**UNIT IV**

1. What are the types of parsers?
2. Types of top down parsers.
3. Types of bottom up parsers.
4. Name a backtracking top down parser.
5. Name a non backtracking top down parser.
6. Why are top down parsers also called LL parsers?
7. List down the bottom up parsers.
8. Why are bottom up parsers call LR parsers?
9. What is handle and handle pruning?
10. Which is the most powerful bottom up( LR) parser?
11. How is the parse tree generated using parser(YACC)?
12. Which type of parser does YACC implement?
13. Which type of grammar does YACC use to generate parse tree?
14. What do you mean by lookahead symbol in parsers?
15. What is meant by LL(k) and LR(k) parsers?
16. Are bottom up parsers backtracking parsers?
17. What type of conflicts are encountered in Shift reduce parsers?

**UNIT V**

1. Define synthesized and inherited attributes.
2. What is an annotated parse tree?
3. List down different representations of Intermediate code generation in parsers.
4. Write three address code(TAC) for

* (a>b ) and (c<d)
* If (a>b) then p=q+r
* While (a>b) do

s=s+1

* for(i=1;i<=10;i++)

j++

1. Write three address code for accessing an element of a single dimensional array and an element of double dimensional array.

**UNIT VI**

1. Which are the types of code optimization?
2. What is

Dead code elimination

Common subexpression elimination

Loop invariant computation.