

# System Software and Compiler Design

- 6.1 A/An \_\_\_\_\_ is a computerized mechanism that transforms source code written in a higher level programming language into a target code, generally in a binary form.
- |                 |                |
|-----------------|----------------|
| (a) Compiler    | (c) Translator |
| (b) Interpreter | (d) Inverter   |
- 6.2 Output of a compiler can be \_\_\_\_\_.
- |   |                    |
|---|--------------------|
| (a) A program of target language                        | (c) Error          |
| (b) Either program in target language or errors, if any | (d) Source program |
- 6.3 A/An \_\_\_\_\_ translates program written in higher level computer language into machine readable code in line by line manner.
- |                 |                |
|-----------------|----------------|
| (a) Interpreter | (c) Translator |
| (b) Compiler    | (d) Editor     |
- 6.4 Parsing in a typical compilation processes is also known as \_\_\_\_\_.
- |                       |                      |
|-----------------------|----------------------|
| (a) Semantic analysis | (c) Semantic editing |
| (b) Syntax analysis   | (d) Interpretation   |
- 6.5 A \_\_\_\_\_ is a data structure used while compilation, to maintain records for each identifier along with necessary information about its fields and attributes.
- |                           |                          |
|---------------------------|--------------------------|
| (a) File allocation table | (c) Memory de-allocation |
| (b) Memory allocation     | (d) Symbol table         |
- 6.6 The last phase of any compiler is \_\_\_\_\_.
- |   |   |
|---|---|
| (a) To generate source code             | (c) To generate code in higher level language |
| (b) To generate code in target language | (d) To optimize code                          |



- 6.7 Pre-processor does \_\_\_\_\_.
- (a) File inclusion
  - (b) Macro substitution
  - (c) Language extension
  - ✓(d) All of these
- 6.8 \_\_\_\_\_ code is a mnemonic version of a binary code.
- (a) Structured query language
  - (b) Higher level programming language
  - (c) Semantic
  - ✓(d) Assembly
- 6.9 \_\_\_\_\_ is the scheme that determines if a string of tokens is generated by given grammar or not.
- (a) Compiling
  - ✓(b) Parsing
  - (c) Editing
  - (d) Linking
- 6.10 In \_\_\_\_\_ parsing mechanism, the construction/ traversal of parse tree starts with root node of the tree and proceed towards leaf nodes.
- (a) Bottom up
  - (b) Sequential
  - ✓(c) Top down
  - (d) Hybrid
- 6.11 A \_\_\_\_\_ performs all its functions by reading the source file once.
- (a) Two -pass assembler
  - ✓(b) One-pass assembler
  - (c) Interpreter
  - (d) Three-pass assembler
- 6.12 Which of the following is not a phase of compilation?
- (a) Lexical analysis
  - (b) Code generation
  - (c) Code optimization
  - ✓(d) Code editing
- 6.13 \_\_\_\_\_ is a program that places programs to be executed into memory and prepares them for execution.
- (a) Parser
  - (b) Memory allocator
  - ✓(c) Loader
  - (d) Interpreter
- 6.14 Compiler can identify \_\_\_\_\_ type of errors.
- (a) Null pointer exception
  - (b) Run time
  - ✓(c) Syntax
  - (d) All type of errors



- 6.15 Instead of an addition operation, if a program uses a multiplication operation and giving unexpected result; such error can be identified by \_\_\_\_\_.  
 (a) Compiler (c) Interpreter  
 - (b) Comparing output with expected output manually (d) Interactive editors
- 6.16 \_\_\_\_\_ is a program that integrates separately compiled modules of a program for smooth execution.  
 (a) Loader - (c) Linker  
 (b) Parser (d) Integrated compiler
- 6.17 \_\_\_\_\_ is an intermediate code in a compiler that can be implemented using quadruples or triples.  
 - (a) Three-address code (c) Infix notation  
 (b) Prefix notation (d) Outfix notation
- 6.18 A \_\_\_\_\_ is a compiler capable of creating executable code for a platform other than the one on which the compiler is running.  
 (a) Advanced compiler (c) Multi compiler  
 (b) Integrated compiler - (d) Cross compiler
- 6.19 LRU is an algorithm used for \_\_\_\_\_ in operating systems.  
 (a) Compaction (c) Disk scheduling  
 - (b) Page replacement (d) CPU scheduling
- 6.20 \_\_\_\_\_ is an assembler that runs on one computer and assembles program for another computer.  
 - (a) Cross assembler (c) Turbo assembler  
 (b) Meta assembler (d) Multi assembler
- 6.21 \_\_\_\_\_ makes the assembler program machine dependent.  
 - (a) Mnemonics operation table (c) Terminal table  
 (b) Macro definition table (d) Symbol table
- 6.22 Forward Reference Table (FRT) is generally implemented as \_\_\_\_\_.  
 (a) Queue (c) Stack  
 - (b) Linked list (d) File



- 6.23 Which of the following will be a part of operating systems and resides in the main memory?  
 (a) Linker  
 (b) Assembler  
 (c) Loader  
 (d) Compiler
- 6.24 Compiler is a kind of \_\_\_\_\_ software.  
 (a) Application systems  
 (b) Operating systems  
 (c) Software engineering  
 (d) System software
- 6.25 Which of the following is not system software?  
 (a) Microsoft Windows  
 (b) Power Point presentation  
 (c) DOS  
 (d) Unix
- 6.26 The initial program, which a computer should start after it's booting (start up) is a / an \_\_\_\_\_.  
 (a) Operating system  
 (b) Loader  
 (c) Compiler  
 (d) Linker
- 6.27 \_\_\_\_\_ is defined as stream of characters to be treated as single unit.  
 (a) Characters  
 (b) Token  
 (c) Instruction  
 (d) Data
- 6.28 Identification of tokens can be done by a \_\_\_\_\_.  
 (a) Deterministic Finite-state automaton  
 (b) Non Deterministic Finite-state automaton  
 (c) Both (a) and (b)  
 (d) DAG
- 6.29 A regular expression is a \_\_\_\_\_.  
 (a) Set of production rules  
 (b) Set of white space  
 (c) Set of instructions  
 (d) None of these
- 6.30 Compiling a newly written compiler program (written in language other than target language) is known as \_\_\_\_\_.  
 (a) Just compiling  
 (b) Bootstrapping  
 (c) De-compilation  
 (d) Cross Compilation



## ANSWERS:

- |      |  |      |   |
|------|--|------|---|
| 6.1  | Compiler                                       | 6.2  | Either program in target language or errors, if any |
| 6.3  | Interpreter                                    | 6.4  | Syntax analysis                                     |
| 6.5  | Symbol table                                   | 6.6  | To generate code in target language                 |
| 6.7  | All of these                                   | 6.8  | Assembly  |
| 6.9  | Parsing  | 6.10 | Top down  |
| 6.11 | One-pass assembler                             | 6.12 | Code editing  |
| 6.13 | Loader   | 6.14 | Syntax  |
| 6.15 | Comparing output with expected output manually | 6.16 | Linker  |
| 6.17 | Three-address code                             | 6.18 | Cross compiler                                      |
| 6.19 | Page replacement                               | 6.20 | Cross assembler                                     |
| 6.21 | Mnemonics operation table                      | 6.22 | Linked list   |
| 6.23 | Loader   | 6.24 | System software                                     |
| 6.25 | Power Point presentation                       | 6.26 | Operating system                                    |
| 6.27 | Token  | 6.28 | Both (a) and (b)                                    |
| 6.29 | None of these                                  | 6.30 | Bootstrapping                                       |