

Question: 1

A compiler which allows only the modified section of the source code to be recompiled is called

- (A) Subjective compiler
- (B) Dynamic compiler
- (C) Re-configurable compiler
- (D) Incremental compiler      Ans: D

Question: 2

A compiler is preferable to an interpreter because

- (A) Debugging can be faster and easier
  - (B) If one changes a statement, only that statement needs re-compilation
  - (C) It is much helpful in the initial stages of program development
  - (D) It can generate stand alone programs that often take less time for execution
- Ans: D

Question: 3

Which of the following symbol table implementation has the minimum access time?

- (A) Self-organizing list
- (B) Linear
- (C) Search tree
- (D) Hash table

Ans: D

Question: 4

A top-down parser generates

- (A) Left-most derivation in reverse
- (B) Left-most derivation
- (C) Right-most derivation in reverse
- (D) Right –most derivation

Ans: B

Question: 5

Which of the following parsers is the most powerful?

- (A) SLR
- (B) LALR

(C) Canonical LR  
(D) Operator-precedence  
Ans: C

Question: 6

Which table is a permanent database that has an entry for each terminal symbol?

- (A) Reductions
- (B) Identifier table
- (C) Literal table
- (D) Terminal table

Ans: D

Question: 7

The action of parsing the source program into proper syntactic classes is called

- (A) General syntax analysis
- (B) Interpretation analysis
- (C) Syntax analysis
- (D) Lexical analysis

Ans: D

**Q8.A bottom-up parser generates**

- (A) Left-most derivation in reverse
- (B) Left-most derivation
- (C) Right-most derivation in reverse
- (D) Right –most derivation

Ans: C

Question: 9

The bottom-up parsing method is also called

- (A) Shift reduce parsing
- (B) Predictive parsing
- (C) Recursive descent parsing
- (D) None of these

Ans: A

Question: 10

The method which merges the bodies of two loops is

- (A) Constant folding
- (B) Loop jamming
- (C) Loop unrolling

(D) None of these

Ans: B

Question: 11

Synthesized attribute can be easily simulated by a

(A) LR grammar

(B) LL grammar

(C) Ambiguous grammar

(D) None of these

Ans: A

Question: 12

The top-down parsing method is also called

(A) Operator precedence parsing

(B) Recursive descent parsing

(C) Shift reduce parsing

(D) None of these

Ans: B

Question: 13

Which of the following techniques is used to replace run-time computations by compile time computations?

(A) Constant folding

(B) Code hoisting

(C) Pee phole optimization

(D) Invariant computation

Ans: A

Question: 14

The most general phase structured grammar

(A) Context sensitive

(B) Context free

(C) Regular

(D) None of these

Ans: A

Question: 15

Which of the following class of statement usually produces no executable code when compiled?

(A) Assignment statement

(B) Structural statements

(C) Input and output statements

(D) Declaration

Ans: B

Question: 16

The lexical analyzer takes \_\_\_\_\_ as input and produces a list of \_\_\_\_\_ of output.

(A) Machine code, mnemonic

(B) Tokens, source code

(C) Source code, tokens

(D) Both a and b

Ans: C

Question: 17

Linear analysis is called \_\_\_\_\_ in a compiler.

(A) Lexical analysis

(B) Scanning

(C) Testing

(D) Both a and b

Ans: D

Question: 18

Lexical analysis is about breaking a sequence of characters into

(A) Tokens

(B) Lines

(C) Groups

(D) Packets

Ans: A

Question: 19

The phase Syntax Analysis is modeled on the basis of

(A) High level language

(B) Low level language

(C) Context free grammar

(D) Regular grammar

Ans: C

Question: 20

Compiler is a program that

(A) Accepts a program written in a high level language and produces an object program

(B) Appears to execute a source program as if it were machine language

(C) Automates the translation of assembly language into machine language

(D) Places programs into memory and prepares them for execution

Ans: A

Question: 21

An optimizing compiler

(A) Is optimized to occupy less space

(B) Optimized the code

(C) Is optimized to take less time for execution

(D) None of these

Ans: B

Question: 22

A compiler for a high level language that runs on one machine and produce code for different machine is called

- (A) One pass compiler
- (B) multipass compiler
- (C) croos compiler
- (D) optimizing compiler

Ans: C

Question: 23

The graph that shows the basic blocks and their successor relationship is called

- (A) Hamiltonian graph
- (B) Control graph
- (C) Flow graph
- (D) DAG

Ans: C

Question: 24

\_\_\_\_\_ constructs the desired target program from the intermediate representation of the source program.

- (A) Analysis part
- (B) Lexical part
- (C) Synthesis part
- (D) None of these

Ans: C

Question: 25

What is the name of the process that determining whether of tokens can be generated by a grammar?

- (A) Analysing
- (B) Parsing

(C) Translating

(D) Recognizing

Ans: B

Question: 26

\_\_\_\_\_ grammars are not phase structured grammar.

(A) Regular

(B) Context free

(C) Context sensitive

(D) None of these

Ans: D

Question: 27

The parsing technique that avoids backtracking is

(A) Top-down parsing

(B) Recursive-descent parsing

(C) Predictive parsing

(D) Both (b) and (c)

Ans: D

Question: 28

CSG (COnText Sensitive Grammar) can be recognized by

(A) Push down automata

(B) Finite state automata

(C) 2-way linear bounded automata

(D) All of the above

Ans: C

Question: 29

Which of the following symbol table implementation is based on the property of locality of reference?

- (A) Linear list
- (B) Self-organizing list
- (C) Search tree
- (D) Hash table

Ans: B

Question: 30

The process of searching for matched tokens is typically described using \_\_\_\_

- (A) Finite automata
- (B) Regular expressions
- (C) Context free grammar
- (D) Both a and b

Ans: D

Question: 31

Loader is a program that

- (A) Places programs into memory and prepares them for execution
- (B) Automates the translation of assembly language into machine language
- (C) Accepts a program written in a high level language and produces an object program
- (D) Appears to execute a source program as if it were machine language

Ans: A

Question: 32

Interpreter is preferred over a compiler is

- (A) During program development phase
- (B) When storage space is to be minimized
- (C) When efficient use of computer resources is the consideration
- (D) All of these

Ans: A



Question: 33

Which of the following is used for grouping of characters into tokens?

- (A) Scanner
- (B) Code generator
- (C) Code optimizer
- (D) Parser

Ans: A

Question: 34

Type checking is normally done during

- (A) Code optimization
- (B) Syntax directed translation
- (C) Lexical analysis
- (D) Syntax analysis

Ans: B

Question: 35

Back- patching is useful for handling

- (A) Forward references
- (B) Backward references
- (C) Conditional jumps
- (D) Unconditional jumps

Ans: A

Question: 36

\_\_\_\_\_ is considered as a sequence of characters in a token.

- (A) Pattern
- (B) Texeme
- (C) Lexeme

(D) Mexeme

Ans: C

Question: 37

\_\_\_\_\_ or scanning is the process, where the stream of characters making up the source program is read from left to right and grouped into tokens.

(A) Modeling

(B) Diversion

(C) Lexical analysis

(D) All of these

Ans: C

Question: 38

LR stands for

(A) Left to right

(B) Left to right reduction

(C) Right to left

(D) Left to right and right most derivation in reverse

Ans: D

Question: 39

Recursive descent parsing belongs to the class of

(A) Top-down parsing

(B) Bottom-up parsing

(C) Predictive parsing

(D) None of these

Ans: A

Question: 40

The graph depicting the inter-dependencies of the attribute of different nodes in a parse is called

- (A) Dependency graph
- (B) Karnaugh's graph
- (C) Steffi graph
- (D) Flow graph

Ans: C

Question: 41

Compiler should report the presence of \_\_\_\_\_ in the source program, in translation process.

- (A) Text
- (B) Errors
- (C) Classes
- (D) Objects

Ans: B

Question: 42

A basic block can be analyzed by

- (A) Graph with cycles
- (B) DAG
- (C) Flow graph
- (D) None of these

Ans: B

Question: 43

Minimum hamming distance method is used for connection of

- (A) Algorithm errors
- (B) Transcription errors
- (C) Semantic errors

(D) Syntactic errors

Ans: D

Question: 44

The output of lexical analyzer is

(A) Strings of characters

(B) A set of tokens

(C) Syntax tree

(D) A set of regular expressions

Ans: B

Question: 45

Symbol table can be used for

(A) Storage allocation

(B) Checking type compatibility

(C) Suppressing duplication of error messages

(D) All of these

Ans: D

Question: 46

Compiler can check \_\_\_\_\_ error.

(A) syntax

(B) logical

(C) content

(D) Both a and b

Ans: A

Question: 47

Which of the following is language processor?

(A) Interpreters

(B) Assembler

(C) Compilers

(D) All of these

Ans: D

Question: 48

**CFG (Context Free Grammar) can be recognized by a**

(A) Push down automata

(B) Finite state automata

(C) 2 way linear bounded automata

(D) Both a and c

Ans: D

Question: 49

**Handle pruning is the technique used to obtain**

(A) Canonical reduction sequence

(B) Canonical derivation sequence

(C) Both (a) and (b)

(D) None of these

Ans: A

Question: 50

**Semantic errors can be detected at**

(A) Compile time only

(B) Run-time only

(C) Both (a) and (b)

(D) None of these

Ans: C

Question: 51

**A permanent database in the general model of compiler is**

(A) Source code

(B) Terminal table

(C) Identifier table

(D) Literal table

Ans: B

Question: 52

Which of the following actions an operator-precedence parser may take to recover from an error?

(A) Insert symbols onto the stack

(B) Delete symbols from the stack

(C) Insert or delete symbols from the input

(D) All of these

Ans: D

Question: 53

Which of the following system software resides in main memory always?

(A) Text editor

(B) Assembler

(C) Linker

(D) All of these

Ans: D

Question: 54

Left factoring is the process of factoring out the common

(A) Prefixes of alternates

(B) Suffixes of alternates

(C) Both(a) and (b)

(D) None of these

Ans: A

Question: 55

\_\_\_\_\_ is/are parts of the compiler.

- (A) Execution part
- (B) Analysis part
- (C) Synthesis part
- (D) Both B and C

Ans: D

Question: 56

How many parts of compiler are there?

- (A) 2
- (B) 4
- (C) 6
- (D) 8

Ans: A

Question: 57

Which of the following is/are the phases of compiler?

- (A) Code generation
- (B) Syntax analyser
- (C) Lexical analyser
- (D) All of these

Ans: D

Question: 58

Compiler translates the source code to

- (A) Machine code
- (B) Executable code
- (C) Binary code
- (D) Both A and C

Ans: D

Question: 59

Grammar of the programming is checked at \_\_\_\_\_ phase of compiler.

- (A) Syntax analysis
- (B) Semantic analysis
- (C) Code generation
- (D) Code optimization

Ans: A

Question: 60

Which of the following groups is/are token together into syntactic structures?

- (A) Syntax analyzer
- (B) Semantic analyzer
- (C) Lexical analyzer
- (D) Intermediate code generation

Ans: C

Question: 61

What is a compiler?

- (A) A compiler is calculating device which is providing very efficient execution
- (B) A compiler is a general purpose language providing very efficient execution
- (C) A compiler converts the whole of a higher level program code into machine code in one step
- (D) A compiler does a conversion line by line as the program is run

Ans: C

Question: 62

Only OS independent compiler is

- (A) Java compiler



(B) Visual basic compiler

(C) Pascal compiler

(D) Turbo C compiler

Ans: A

Question: 63

The phase 'Semantic Analysis' is responsible for \_\_\_\_\_ in Compiler.

(A) Check semantics

(B) Static checking

(C) Type checking

(D) All of these

Ans: D

Question: 64

The errors that can be pointed out by the compiler are known as

(A) Internal errors

(B) Logical errors

(C) Semantic errors

(D) Syntax errors

Ans: D

Question: 65

What do you call the translator which takes assembly language program as input & produce machine language code as output?

(A) Assembler

(B) Compiler

(C) Debugger

(D) Interpreter

Ans: A

Question: 66

Which of the following cannot be used as an intermediate code form?

- (A) Quadruples
- (B) Syntax trees
- (C) Three address codes
- (D) Post fix notation

Ans: A

Question: 67

Which of the following can be used to identify loops?

- (A) Dominators
- (B) Reducible graphs
- (C) Depth first ordering
- (D) All of these

Ans: D

Question: 68

The cost of developing a compiler is proportional to

- (A) Flexibility of the available instruction set
- (B) Complexity of the architecture of the target machine
- (C) Complexity of the source language
- (D) All of these

Ans: D

Question: 69

Replacement of an expensive operation by a cheaper one is called

- (A) Reduction in strength
- (B) Loop-invariant computation

(C) Code motion

(D) None of these

Ans: A

Question: 70

Which of the following is not a source of error?

(A) Faulty design specification

(B) Faulty algorithm

(C) Compilers themselves

(D) None of these

Ans: D

Question: 71

Programming languages can be categories as \_\_\_\_\_

(A) Assembly language

(B) High level language

(C) Machine language

(D) All of these

Ans: D

Question: 72

What is the meaning of a Compiler?

(A) Name given to the computer operator

(B) Part of the digital machine to store information

(C) Operator of Boolean algebra

(D) Translator to convert source program into object code

Ans: D

Question: 73

Language used in a computer that is similar to the languages of human and is easy to understand, is referred as

- (A) Object code
- (B) Source code
- (C) Machine language
- (D) High level language

Ans: D

Question: 74

Which among the following is/are interpreted language?

- (A) C++
- (B) Java
- (C) Visual basic
- (D) Both B and C

Ans: D

Question: 75

What is a notation for writing programs, which are specifications of a computation or algorithm?

- (A) A programming language
- (B) An operating system
- (C) Hardware
- (D) Web browser

Ans: A

Question: 76

What is the action of parsing the source program into proper syntactic classes?

- (A) Syntax analysis
- (B) Lexical analysis
- (C) Interpretation analysis
- (D) General syntax analysis

Ans: B

Question: 77

\_\_\_\_\_ is a process of finding a parse tree for a string of tokens.

- (A) Analysing
- (B) Recognizing
- (C) Parsing
- (D) Tokenizing

Ans: C

Question: 78

Representing the syntax by a grammar is advantageous. What is the cause?

- (A) It is concise
- (B) It is accurate
- (C) Automation becomes easy
- (D) All of the above

Ans: D

Question: 79

A grammar that produces more than one parse tree for some sentence is called as

- (A) Ambiguous
- (B) Regular
- (C) Unambiguous
- (D) All of these

Ans: A

Question: 80

A \_\_\_\_\_ is a software utility that translates code written in higher language into a low level language.

- (A) Compiler
- (B) Converter
- (C) Text editor
- (D) Code optimizer

Ans: A

Question: 81

Which programming languages are classified as low level languages?

- (A) BASIC, COBOL, FORTRAN
- (B) Assembly languages
- (C) Knowledge based Systems
- (D) Prolog 2, Expert Systems

Ans: B

Question: 82

A compiler translate high level programs into low level programs, which is called as

- (A) Beta code
- (B) Compiled code
- (C) Source code
- (D) Object code

Ans: D

Question: 83

What is an interpreter?

- (A) An interpreter is a decoder which provides very efficient execution
- (B) An interpreter is a general purpose language providing very efficient execution
- (C) An interpreter does the conversion line by line as the program is run
- (D) An interpreter is the representation of the system being designed

Ans: D

Question: 84

The translator program used in assembly language is called

- (A) Assembler
- (B) Interpreter
- (C) Compiler
- (D) Translator

Ans: A

Question: 85

A(n) \_\_\_\_\_ language reflects the way people think mathematically.

- (A) Functional programming
- (B) Cross platform programming
- (C) 3 GL business programming
- (D) Event driven programming

Ans : A

86. Pee Hole optimization \_\_\_\_\_

- a) Loop Optimization
- b) Local Optimization
- c) Constant folding
- d) Data Flow analysis

Answer: c

87. The optimization which avoids test at every iteration is?

- a) Loop unrolling
- b) Loop jamming
- c) Constant folding
- d) None of the mentioned

Answer: a

88. Shift reduce parsers are \_\_\_\_\_

- a) Top down Parser
- b) Bottom Up parser
- c) May be top down or bottom up
- d) None of the mentioned

Answer: b

1. DAG representation of a basic block allows \_\_\_\_\_
  - a) Automatic detection of local common sub expressions
  - b) Detection of induction variables
  - c) Automatic detection of loop variant
  - d) None of the mentionedAnswer: a
2. Inherited attribute is a natural choice in \_\_\_\_\_
  - a) Tracking declaration of a variable
  - b) Correct use of L and R values
  - c) All of the mentioned
  - d) None of the mentionedAnswer: a
3. An intermediate code form is \_\_\_\_\_
  - a) Postfix notation
  - b) Syntax Trees
  - c) Three Address code
  - d) All of the mentionedAnswer: d
4. Which of the following actions an operator precedence parser may take to recover from an error?
  - a) Insert symbols onto the stack
  - b) Delete symbols from the stack
  - c) Inserting or deleting symbols from the input
  - d) All of the mentionedAnswer: d
5. What is the output of lexical analyzer?
  - a) A set of regular expression
  - b) Syntax trees
  - c) Set of Token
  - d) String of CharactersAnswer: c
6. Which of the following is used for grouping of characters into tokens?
  - a) Parser
  - b) Code optimization
  - c) Code generator
  - d) Lexical analyserAnswer: d
7. Shift reduce parsers are \_\_\_\_\_
  - a) Top down parser
  - b) Bottom up parser
  - c) Maybe both
  - d) None of the mentionedAnswer: b
8. A bottom up parser generates \_\_\_\_\_
  - a) Right most derivation
  - b) Right most derivation in reverse
  - c) Left most derivation
  - d) Left most derivation in reverseAnswer: b
9. What is garbage?
  - a) Unallocated storage
  - b) Allocated storage whose access paths are destroyed?
  - c) Allocated storage
  - d) Uninitialized storageAnswer: b
10. An optimizing compiler \_\_\_\_\_
  - a) Is optimized to occupy less space
  - b) Is optimized to take less time for execution
  - c) Optimized the code
  - d) None of the mentionedAnswer: c



11. Input to code generator is \_\_\_\_\_  
a) Source code  
b) Intermediate code  
c) Target code  
d) All of the mentioned Answer: b
12. A synthesized attribute is an attribute whose value at a parse tree node depends on \_\_\_\_\_  
a) Attributes at the siblings only  
b) Attributes at parent node only  
c) Attributes at children nodes only  
d) None of the mentioned Answer: c
13. In a bottom up evaluation of a syntax direction definition, inherited attributes can \_\_\_\_\_  
a) Always be evaluated  
b) Be evaluated only if the definition is L –attributed  
c) Evaluation only done if the definition has synthesized attributes  
d) None of the mentioned Answer: c
14. The graph that shows basic blocks and their successor relationship is called \_\_\_\_\_  
a) DAG  
b) Flow Chart  
c) Control Graph  
d) Hamilton graph Answer: b
15. \_\_\_\_\_ or scanning is the process where the stream of characters making up the source program is read from left to right and grouped into tokens.  
a) Lexical Analysis  
b) Diversion  
c) Modelling  
d) None of the mentioned Answer: a
16. \_\_\_\_\_ is a graph representation of a derivation.  
a) The parse tree  
b) Oct tree  
c) Binary tree  
d) None of the mentioned Answer: a
17. Assume that the SLR parser for a grammar G has  $n_1$  states and the LALR parser for G has  $n_2$  states. Hence which one is true?  
a)  $N_1$  is necessarily less than  $n_2$   
b)  $N_1$  is necessarily equal to  $n_2$   
c)  $N_1$  is necessarily greater than  $n_2$   
d) None of the mentioned Answer: b
18. Which of these is also known as look-head LR parser?  
a) SLR  
b) LR  
c) LLR  
d) None of the mentioned Answer: c
19. What is the similarity between LR, LALR and SLR?  
a) Use same algorithm, but different parsing table  
b) Same parsing table, but different algorithm  
c) Their Parsing tables and algorithm are similar but uses top down approach  
d) Both Parsing tables and algorithm are different Answer: a
20. An LR-parser can detect a syntactic error as soon as \_\_\_\_\_  
a) The parsing starts  
b) It is possible to do so a left-to-right scan of the input  
c) It is possible to do so a right-to-left scan of the input  
d) Parsing ends Answer: b

21. Which of these is true about LR parsing?
- Is most general non-backtracking shift-reduce parsing
  - It is still efficient
  - Is most general non-backtracking shift-reduce parsing & It is still efficient
  - None of the mentioned
- Answer: c
22. Which of the following is incorrect for the actions of A LR-Parser I) shift s ii) reduce A-> $\beta$  iii) Accept iv) reject?
- Only I)
  - I) and ii)
  - I), ii) and iii)
  - I), ii) , iii) and iv)
- Answer: c
23. If a state does not know whether it will make a shift operation or reduction for a terminal is called \_\_\_\_\_
- Shift/reduce conflict
  - Reduce /shift conflict
  - Shift conflict
  - Reduce conflict
- Answer: a
24. When there is a reduce/reduce conflict?
- If a state does not know whether it will make a shift operation using the production rule i or j for a terminal
  - If a state does not know whether it will make a shift or reduction operation using the production rule i or j for a terminal
  - If a state does not know whether it will make a reduction operation using the production rule i or j for a terminal
  - None of the mentioned
- Answer: c
25. When  $\beta$  (in the LR(1) item A ->  $\beta.a,a$ ) is not empty, the look-head \_\_\_\_\_
- Will be affecting
  - Does not have any affect
  - Shift will take place
  - Reduction will take place
- Answer: b
26. When  $\beta$  is empty (A ->  $\beta.,a$ ), the reduction by A-> a is done by \_\_\_\_\_
- If next symbol is a terminal
  - Only If the next input symbol is a
  - Only If the next input symbol is A
  - Only if the next input symbol is a
- Answer: d
27. The construction of the canonical collection of the sets of LR (1) items are similar to the construction of the canonical collection of the sets of LR (0) items. Which is an exception?
- Closure and goto operations work a little bit different
  - Closure and goto operations work similarly
  - Closure and additive operations work a little bit different
  - Closure and associatively operations work a little bit different
- Answer: a
28. What is terminal table?
- Contains all constants in the program
  - Is a permanent table of decision rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure
  - Consist of a full or partial list of the token is as they appear in the program created by lexical analysis and used for syntax analysis and interpretation
  - Is a permanent table which lists all keywords and special symbols of the language in symbolic form
- Answer: d
29. Advantage of incorporating the macro-processor into pass 1 is that \_\_\_\_\_
- Many functions have to be implemented twice
  - Functions are combined not necessarily creating intermediate files as output from the macro-

processor and input to the assembler

c) More flexibility is provided to the programmer in that he may use all the features of the assembler in conjunction with macros

d) All of the mentioned Answer: d

**30.** Which of the following is a phase of a compilation process?

a) Lexical Analysis

b) Code Generation

c) Lexical Analysis & Code Generation

d) None of the mentioned Answer: c

**31.** System program such as compiler are designed so that they are \_\_\_\_\_

a) Re-enterable

b) Non reusable

c) Serially usable

d) None of the mentioned Answer: a

**32.** A series of statements explaining how the data is to be processed is called \_\_\_\_\_

a) Assembly

b) Machine

c) COBOL

d) Program Answer: d

**33.** A loader is a program that \_\_\_\_\_

a) Program that places functions into memory and prepares them for execution

b) Program that automates the translation of assembly language into machine language

c) Program accepting another program written in a high level language and produces as object program

d) None of the mentioned Answer: a

**34.** A system program that setup an executable program in main memory ready for execution is?

a) Assembler

b) Linker

c) Loader

d) Load and go Answer: c

**35.** Which of the following system program forgoes the production of object code to generate absolute machine code and load it into the physical main storage location from which it will be executed immediately upon completion of the assembly?

a) Two pass assembler

b) Load and go assembler

c) Macro processor

d) Linker Answer: b

**36.** Uniform symbol table \_\_\_\_\_

a) Has all constants in the program

b) Permanent table of rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure

c) Consists of full or partial list of the tokens as they appear in the program created by Lexical analysis and used for syntax analysis and interpretation

d) A permanent table which has all key words and special symbols of the language in symbolic form Answer: c

**37.** Assembler is a program that \_\_\_\_\_

a) Puts programs into memory and executes them

b) Translates the assembly language into machine language

c) Writes in high level language and produces an object program

d) None of the mentioned Answer: b

**38.** A simple two-pass assembler does which of the following in the first pass?

a) It allocates space for the literals

- b) Calculates total length of the program
  - c) Symbol table is built for the symbols and their value
  - d) All of the mentioned Answer: d
- 39.** A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar  $S \rightarrow xxW$  ( PRINT "1" )  $S \rightarrow y \{ \text{print " 2 " } \}$   $S \rightarrow Sz \{ \text{print " 3 " } \}$  ) What is the translation of xxxxyzz using the syntax directed translation scheme described by the above rules?
- a) 23131
  - b) 11233
  - c) 11231
  - d) 33211 Answer: a
- 40.** In operator precedence parsing whose precedence relations are defined \_\_\_\_\_
- a) For all pair of non-terminals
  - b) For all pair of terminals
  - c) To delimit the handle
  - d) None of the mentioned Answer: a
- 41.** LR parsers are attractive because \_\_\_\_\_
- a) It can be constructed to recognize CFG corresponding to almost all programming constructs
  - b) It does not backtrack
  - c) It can be constructed to recognize CFG corresponding to almost all programming constructs & It does not backtrack
  - d) None of the mentioned Answer: c
- 42.** Which is the most powerful parser?
- a) SLR
  - b) LALR
  - c) Canonical LR
  - d) Operator Precedence Answer: c
- 43.** The address code involves \_\_\_\_\_
- a) Exactly 3 address
  - b) At most Three address
  - c) No unary operators
  - d) None of the mentioned Answer: d
- 44.** An intermediate code form is \_\_\_\_\_
- a) Postfix Notation
  - b) Syntax Trees
  - c) Three address code
  - d) All of the mentioned Answer: d
- 45.** In a single pass assembler, most of the forward references can be avoided by putting the restriction \_\_\_\_\_
- a) On the number of strings/life reacts
  - b) Code segment to be defined after data segment
  - c) On unconditional rump
  - d) None of the mentioned Answer: b
- 46.** Assembly code data base is associated with \_\_\_\_\_
- a) Code is converted into assembly
  - b) Table of rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure
  - c) All of the mentioned
  - d) None of the mentioned Answer: a
- 47.** What is the function of the syntax phase?
- a) recognize the language and to cal the appropriate action routines that will generate the intermediate form or matrix for these constructs

- b) Build a literal table and an identifier table
  - c) Build a uniform symbol table
  - d) Parse the source program into the basic elements or tokens of the language Answer: a
- 48.** If  $E$  be a shifting operation applied to a function  $f$ , such that  $E(f) = f(x + \beta)$ , then?
- a)  $E(\alpha f + \beta g) = \alpha E(f) + \beta E(g)$
  - b)  $E(\alpha f + \beta g) = (\alpha + \beta) + E(f + g)$
  - c)  $E(\alpha f + \beta g) = \alpha E(f + g\beta)$
  - d)  $E(\alpha f + \beta g) = \alpha\beta E(f + g)$  Answer: a
- 49.** Pass I \_\_\_\_\_
- a) Assign address to all statements
  - b) Save the values assigned to all labels for use in pass 2
  - c) Perform some processing
  - d) All of the mentioned Answer: d
- 50.** Which table is a permanent database that has an entry for each terminal symbol?
- a) Terminal Table
  - b) Literal Table
  - c) Identifier Table
  - d) None of the mentioned Answer: a
- 51.** The root directory of a disk should be placed \_\_\_\_\_
- a) At a fixed address in main memory
  - b) At a fixed location on the disk
  - c) Anywhere on the disk
  - d) None of the mentioned Answer: b
- 52.** The segment base is specified using the register named is?
- a) ORG instructions
  - b) TITLE instruction
  - c) ASSUME instruction
  - d) SEGMENT instruction Answer: a
- 53.** In what module multiple instances of execution will yield the same result even if one instance has not terminated before the next one has begun?
- a) Non usable module
  - b) Serially usable
  - c) Re-enter-able
  - d) None of the mentioned Answer: c
- 54.** Which of the following is true for machine language?
- a) Continuous execution of program segments
  - b) Depicting flow of data in a system
  - c) A sequence of instructions which solves a problem
  - d) The language which interacts with the computer using only the binary digits 1 and 0 Answer: d
- 55.** The table created by lexical analysis to describe all literals used in the source program is?
- a) Terminal table
  - b) Literal table
  - c) Identifier table
  - d) Reductions Answer: b
- 56.** Which is not true about syntax and semantic parts of a computer language?
- a) Semantics is checked mechanically by a computer
  - b) Semantics is the responsibility of the programmer
  - c) All of the mentioned
  - d) None of the mentioned Answer: d
- 57.** Which of the following statement is true?
- a) SLR powerful than LALR
  - b) LALR powerful than Canonical LR parser

- c) Canonical LR powerful than LALR parser
  - d) The parsers SLR= Canonical LR=LALR Answer: c
- 58.** Which of the following features cannot be captured by CFG?
- a) Syntax of if-then-else statements
  - b) Syntax of recursive procedures
  - c) A variable is declared before its use
  - d) Matching nested parenthesis Answer: d
- 59.** In which of the following no information hiding is done?
- a) Compile prog 1, prog 2
  - b) Run test, prog
  - c) Load R1, A
  - d) 001001000010101 Answer: d
- 60.** The identification of common sub-expression and replacement of run-time computations by compile-time computations is \_\_\_\_\_
- a) Local optimization
  - b) Loop optimization
  - c) Constant folding
  - d) Data flow analysis Answer: c
- 61.** The specific task storage manager performs \_\_\_\_\_
- a) Allocation/ deal location of programs
  - b) Protection of storage area assigned to the program
  - c) Allocation/ deal location of programs & protection of storage area assigned to the program
  - d) None of the mentioned Answer: c
- 62.** When a computer is rebooted, a special type of loader is executed called?
- a) Compile and GO " loader
  - b) Boot loader
  - c) Bootstrap Loader
  - d) Relating Loader Answer: c
- 63.** What is the disadvantage of " Compile and GO " loading scheme?
- a) Memory is wasted because the case occupied by the assembler is unavailable to the object program
  - b) Necessary to translate the users program
  - c) It is very difficult to handle multiple segments, even when the source programs are in different languages and to produce orderly modular programs
  - d) All of the mentioned Answer: d
- 64.** What is the function of the storage assignment?
- a) Assign storage to all variables referenced in the source program
  - b) Assign storage to all temporary locations that are necessary for intermediate results
  - c) Assign storage to literals, and to ensure that the storage is allocated and appropriate locations are initialized
  - d) All of the mentioned Answer: d
- 65.** Generation of intermediate code based on a abstract machine model is useful in compilers because \_\_\_\_\_
- a) Implementation of lexical analysis and syntax analysis is made easier
  - b) Writing for intermediate code generation
  - c) Portability of the front end of the compiler
  - d) None of the mentioned Answer: a
- 66.** Which of the following module does not incorporate initialization of values changed by the module?
- a) Non reusable module
  - b) Serially reusable module

- c) Re-enterable module
- d) All of the mentioned Answer: a

67. The best way to compare the different implementations of symbol table is to compare the time required to \_\_\_\_\_

- a) Add a new name
- b) Make an enquiry
- c) Add a new name and make an enquiry
- d) All of the mentioned Answer: d

68. Dynamic linking can cause security concerns because \_\_\_\_\_

- a) Security is dynamic
- b) None of the mentioned
- c) Security is dynamic & Cryptographic procedures are not available for dynamic linking
- d) Cryptographic procedures are not available for dynamic linking Answer: d

69. Which of the following statements is FALSE?

- a) In up-typed languages, values do not have any types & Dynamic languages have no variables types
- b) In up-typed languages, values do not have any types
- c) Dynamic languages have no variables types
- d) None of the mentioned Answer: c

70. What is grammar?

71.  $S \rightarrow CC$

72.  $C \rightarrow eC \mid d$

73. a) LL(1)

- b) SLR(1) but not LL(1)
- c) LALR(1) but not SLR(1)
- d) LR (1) but not LALR (1) Answer: c

74. The 3-address code sequence generated by which definition?

```
75. S -> id: = E "
76. newtemp ();
77. gen(t . place . place;); .place t} "
78. .place .place;}
79. For the statement 'X: = Y + Z ',
```

- 80. a)  $X=Y+Z$
- b)  $t1=Y+Z; X=t1$
- c)  $t1=Y; t2=t1+Z; X=t2$
- d)  $t1 = Y; t2 = Z; t3 = t2; X = t3$  Answer: d

81. Which of the following is NOT an advantage of using shared, dynamically linked libraries as compared to statically linked libraries?

- a) Smaller sizes of executable
- b) Lesser overall page fault rate in the system
- c) Faster program start-up
- d) Existing programs need not be re-linked to take advantage of newer versions of libraries Answer: c

82. Consider the grammar rule.  $E \rightarrow E1 - E2$  for arithmetic expressions. If  $E1$  and  $E2$  do not have any common sub expression, in order to get the shortest possible code.

- a) E1 should be evaluated first
- b) E2 should be evaluated first
- c) Evaluation of E1 and E2 should necessarily be interleaved
- d) Order of evaluation of E1 and E2 is of no consequence Answer: b

83. The grammar  $A \rightarrow AA \mid (A) \mid \epsilon$  is not suitable for predictive-parsing because the grammar is

- a) Ambiguous
- b) Left-recursive
- c) Right-recursive
- d) An operator-grammar Answer: a

84. Consider the grammar.

85.  $E \rightarrow E + n \mid E \# n \mid n$

86. For a sentence  $n+n$ , form of the reduction are

<ul style="list-style-type: none"> <li>a) <math>n, E+n</math></li> <li>b) <math>n, E+n</math></li> <li>c) <math>n, n+n</math></li> <li>d) <math>n, E+n</math> and <math>E \# n</math></li> </ul>	<div style="border-top: 1px solid black; margin-bottom: 5px;"></div> <ul style="list-style-type: none"> <li>and <math>E+n\#n</math></li> <li>and <math>E+E\#n</math></li> <li>and <math>n+n\#n</math></li> </ul>
--	--

Answer: d

87.  $S \rightarrow (S) \mid a$

Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar  $n_1$ ,  $n_2$  and  $n_3$  respectively.

- a)  $n_1 < n_2 < n_3$
- b)  $n_1 = n_3 < n_2$
- c)  $n_1 = n_2 = n_3$
- d)  $n_1 \neq n_3 \neq n_2$  Answer: b

88. Which one of the following is true about the action of yacc for the given grammar?

- a) It detects recursion and eliminates recursion
  - b) It detects reduce-reduce conflict and resolves
  - c) It detects shift-reduce conflict and resolves the conflict in favor of a shift over a reduce action
  - d) It detects shift-reduce conflict and resolves the conflict in favor of a reduce over a shift action
- Answer: c

89. What precedence and associativity properties does the generated parser realize?

- a) Equal precedence and left associativity; expression is evaluated to 7
- b) Equal precedence and right associativity, expression is evaluated to 9
- c) Precedence of 'x' is higher than that of '+', and both operators are left associative; expression is evaluated to 7
- d) Precedence of '#' is higher than that of '+', and both operators are left associative; expression is evaluated to 9 Answer: b

90. Consider the following grammar.

```

91. S > S * E
92. S > E
93. E > F + E
94. E > F
95. F > id

```

96. Consider the following LR(0) items corresponding to the grammar above.

```

97. (i) S -> S * .E
98. (ii) E -> F . + E

```



99. (iii)  $E \rightarrow F + .E$

100. Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?
- a) (i) and (ii)
  - b) (ii) and (iii)
  - c) (i) and (iii)
  - d) None of the mentioned
- Answer: c

101. Consider the following grammar:

102.  $S \rightarrow FR$

103.  $R \rightarrow * S \mid \epsilon$

104.  $F \rightarrow id$

105. In the predictive parser table,  $M$ , of the grammar the entries  $M[S, id]$  and  $M[R, \$]$  respectively.
- a)  $\{S"FR\}$  and  $\{R"\epsilon\}$
  - b)  $\{S"FR\}$  and  $\{\}$
  - c)  $\{S"FR\}$  and  $\{R"*S\}$
  - d)  $\{F"id\}$  and  $\{R"\epsilon\}$
- Answer: a

106. Consider the following translation scheme.

107.  $S \rightarrow ER$

108.  $R \rightarrow * E \{ \text{print}(' * ');$

109.  $R \mid f$

110.  $E \rightarrow F + E \{ \text{print}(' + '); \mid F F \rightarrow (S) \mid id \{ \text{print}(id.value); \}$

111. Here  $id$  is a token that represents an integer and  $id.value$  represents the corresponding integer value. For an input '2 \* 3 + 4', this translation scheme prints?
- a) 2\*3+4
  - b) 2\*+34
  - c) 23\*4+
  - d) 2 3 4 + \*
- Answer: b

112. Consider the following C code segment.

113. `for (i = 1; i <= 10; i++)`

114. Which one of the following is false?

- a) The code contains loop-invariant computation
  - b) There is scope for common sub-expression elimination in this code
  - c) There is scope for strength reduction in this code
  - d) There is scope for dead code elimination in this code
- Answer: d

115. Consider the following grammar.

116.  $S \rightarrow S * E$

117.  $S \rightarrow E$

118.  $E \rightarrow F + E$

119.  $E \rightarrow F$

120.  $F \rightarrow id$

121. Consider the following LR (0) items corresponding to the grammar above.

122. (i)  $S \rightarrow S * .E$

123. (ii)  $E \rightarrow F . + E$

124. (iii)  $E \rightarrow "F + .E$

125. Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?

- a) (ii)
- b) (i) and (iii)
- c) (iii)
- d) None of the mentioned Answer: C

Which one of the following grammars generates the language  $L = \{a^i b^j \mid i \neq j\}$ ?

a)  $S \rightarrow AC \mid CB$

b)

$S \rightarrow aS \mid Sb \mid a \mid b$   
 $C \rightarrow aCb \mid a \mid b$   
 $A \rightarrow aA \mid \epsilon$   
 $B \rightarrow Bb \mid \epsilon$

c)

d)  $S \rightarrow ACCB$

d)

$S \rightarrow AC \mid CB$   
 $C \rightarrow aCb \mid !$   
 $C \rightarrow aCb \mid !$   
 $A \rightarrow aA \mid !$   
 $A \rightarrow aA \mid a$   
 $B \rightarrow Bb \mid !$   
 $B \rightarrow bB \mid b$

Answer: d

126. Which one of the following is a top-down parser?

- a) Recursive descent parser
- b) Operator precedence parser
- c) An LR(k) parser
- d) An LALR(k) parser Answer: a

127. Consider the grammar with non-terminals.

$N = \{S, C, S\}$ , terminals  $T = \{a, b, i, t, e\}$ , with S as the start symbol, and the following of rules  
 $S \rightarrow iCtSS1 \mid a$   
 $S1 \rightarrow eS \mid \epsilon$   
 $C \rightarrow b$

The grammar is NOTLL(1) because \_\_\_\_\_

- a) It is left recursive
- b) It is right recursive
- c) It is ambiguous
- d) It is not context-free Answer: a

128. Consider the following two statements:

P: Every regular grammar is LL(1)

Q: Every regular set has LR(1) grammar

Which of the following is TRUE?

- a) Both P and Q are true
  - b) P is true and Q is false
  - c) P is false and Q is true
  - d) Both P and Q are false
- Answer: a

129. Which of the following strings is generated by the grammar?

$S \rightarrow bA$	$S \rightarrow aB$
$A \rightarrow a$	$B \rightarrow b$
$A \rightarrow aS$	$B \rightarrow bS$
$A \rightarrow bAA$	$B \rightarrow aBB$

- a) aaaabb
  - b) aabbbb
  - c) aabbab
  - d) abbbba
- Answer: c

130. How many derivation trees are there?

$S \rightarrow bA$	$S \rightarrow aB$
$A \rightarrow a$	$B \rightarrow b$
$A \rightarrow aS$	$B \rightarrow bS$
$A \rightarrow bAA$	$B \rightarrow aBB$

- a) 1
  - b) 2
  - c) 3
  - d) 4
- Answer: b

131. Which of the following describes a handle (as applicable to LR-parsing) appropriately?

- a) It is the position in a sentential form where the next shift or reduce operation will occur
  - b) It is a non-terminal whose production will be used for reduction in the next step
  - c) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.
  - d) It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found
- Answer: d

132. Some code optimizations are carried out on the intermediate code because \_\_\_\_\_

- a) They enhance the portability of the compiler to other target processors
  - b) Program analysis is more accurate on intermediate code than on machine code
  - c) The information from data flow analysis cannot otherwise be used for optimization
  - d) The information from the front end cannot otherwise be used for optimization
- Answer: b

133. An LALR(1) parser for a grammar can have shift-reduce (S-R) conflicts if and only if \_\_\_\_\_

- a) The SLR(1) parser for G has S-R conflicts
  - b) The LR(1) parser for G has S-R conflicts
  - c) The LR(0) parser for G has S-R conflicts
  - d) The LALR(1) parser for G has reduce-reduce conflicts
- Answer: b

134. The lexical analyzer takes \_\_\_\_\_ as input and produces a stream of \_\_\_\_\_ as output.

- a) Source program, tokens
  - b) Token, source program
  - c) Either A and B
  - d) None of the mentioned
- Answer: a

135. The action of parsing the source program into proper syntactic classes is called \_\_\_\_\_

- a) Syntax analysis
  - b) Lexical analysis
  - c) Interpretation analysis
  - d) General syntax analysis
- Answer: b

136. What is the task of the lexical analysis?

- a) None of the mentioned
  - b) To build a literal and identifier table
  - c) To build a uniform symbol table
  - d) To build a uniform symbol table, literal and identifier table
- Answer: d