SIDDHARTH GROUP OF INSTITUTIONS :: PUTTUR
Siddharth Nagar, Narayanavanam Road – 517583

QUESTION BANK (DESCRIPTIVE)

Subject with Code : Compiler Design (13A05502) **Course & Branch :** B. Tech - CSE

Year & Sem : III B.Tech & I-Sem **Regulation :** R13

UNIT-1

1.Explain the need for dividing the compilation process into various phases and explain itsfunctions. Explain how abstract stack machine can be used as translators. [L2, 10M]

2. a) Describe the role performed by lexical analysis of the compiler [L1, 5M]
   b) Explain the need of code optimization in compiler [L2, 5M]
3. Explain programming language basics in detail. [L2, 10M]
4. a) Explain Symbol table management and error handling technique in compiler [L2, 5M]
   b) Differentiate between compiler and interpreter. [L4, 5M]
5. a)Explain construction tools in compiler [L2, 5M]
   b) Explain the different Language processor of a program. [L2, 5M]
6. Define LEX. Explain the use and form of lex program with an example. [L1, 10M]
7. Write short notes [L6, 5+5M]
   a) pass and phases of a compiler
   b) bootstrapping
8. Explain briefly how to recognize tokens in lexical analysis. [L2, 10M]
9. Write short notes [L6, 5+5M]
   a) Application of compiler technology
   b) Parts of compiler
10. a)List the various phases of a compiler. [L1, 2M]
    b)Differentiate tokens, patterns, lexeme. [L4, 2M]
    c) List the operations on languages. [L1, 2M]
    d)Define Regular Expressions and Regular Grammar. [L1, 2M]
    e) List the various error recovery strategies for a lexical analysis. [L1, 2M]

UNIT-2

1.a) Construct the recursive decent parser for the string id*(id+id) following grammar? [L4, 5M]

E-> E+T/T
T-> T*F/F
F-> (E)/id

b) Explain about Left factoring with an example? [L2, 5M]

2. Define augmented grammar? Construct the LR(0) items for the following Grammar? [L1, 10M]

S->L=R
S->R
L->*R

L->id
R->L

3. Calculate first and follow for the following grammar? [L3, 5M]

a) E-> E+T/T
T-> T*F/F
F-> (E)/id

b) S->xABC [L3, 5M]
A->a|bbD
B->a|ε
C->b| ε
D->c| ε

4. Consider the grammar E->E+T,T->T*F,F->(E)|id. Using predictive parsing table parse the string id+id*id. [L3, 10M]

5. Perform Shift Reduce Parsing for the input string (id*id+id) fr the following. [L3, 10M]
E-> E+E|E*E|(E)|id

6. a) For the given grammar S->cAd , A->ab|a , draw the parser tree for the input string w=cad using recursive descent parsing with backtracking. [L4, 5M]

b) For the given grammar T->dFa F->bg|b , draw the parser tree for the input string w=dba using recursive descent parsing with backtracking. [L4, 5M]

7. Consider the grammar
S->AB|ABad
A->d
E ->b
D->b| ε
B->c

Derive the predictive parsing table. Show that the given grammar is LL(1) or not [L3, 10M]

8. Consider the grammar S->xABC
A->a|bbD
B->a|ε
C->b| ε
D->c| ε
Derive the predictive parsing table. [L3, 10M]

9. Perform Shift Reduce Parsing for the input string using the grammar. [L4, 5+5M]
S->(L)|a
L->L,S|S
a) (a,(a,a))
b) (a,a)

10 a) What is phrase level error recovery? [L1, 2M]
b) What are the different strategies of error recovery?. [L1, 2M]
c) Define Left factoring. [L1, 2M]
d) What is Shift – Reduce parsing?. [L1, 2M]
e) What is ambiguous grammar?Give an example. [L2, 2M]

UNIT-3

1. Explain syntax directed definition. [L2, 10M]
2. Describe the evaluation order of SDT with an example. [L5, 10M]
3. Explain the type expression and type equivalence. [L2, 10M]
4. Explain the Translation scheme of SDD. [L2, 10M]
5. Describe the different representation of 3-address code with an example. [L5, 10M]
6. Explain in detail about Backpatching Techniques?. [L2, 10M]
7. Explain the applications of Syntax Directed Definition. [L2, 10M]
8. Write down the translation procedure for control statement and switch statement [L6, 10M]
9. Explain different types of intermediate code with an example. [L2, 10M]
10. a) Define a syntax-directed translation. [L1, 2M]
    b) Define annotated parse tree. [L1, 2M]
    c) What are the three functions of backpatching? [L1, 2M]
    d) Write the Syntax of case statement?. [L6, 2M]
    e) Differentiate between L attribute and S attribute. [L4, 2M]

UNIT-4

1. Draw the format of Activation Record in stack allocation and explain each field in it. [L4, 10M]
2. Explain about Induction variable & Global data flow analysis. [L2, 10M]
3. Explain about the loop optimization techniques with an example. [L2, 10M]
4. Define Symbol table. Explain different types of Data structure for symbol table [L1, 10M]
5. Distinguish between static scope and dynamic scope. Briefly explain access to non-local names in static scope. [L4, 10M]
6. Explain the basic principles source of optimization. [L2, 10M]
7. Explain basic concept of static and dynamic storage allocation. [L2, 10M]
8. Explain heap management mechanism. [L2, 10M]
9. Write briefly reference counting garbage collectors. [L6, 10M]
10. a) Write any four algebraic simplification [L6, 2M]
    b) Name any four procedural optimization techniques [L6, 2M]
    c) Define scope and life time of variable. [L1, 2M]
    d) Define symbol table. [L1, 2M]
    e) What is meant by data flow equation?. [L1, 2M]

UNIT-5

1. Write about all issues in code generation. Describe it. [L6, 10M]
2. Explain the target machine architecture? [L2, 10M]
3. Write about code scheduling. [L6, 10M]
4. Describe the various strategies in register allocation. [L5, 10M]

5. Explain the peephole optimization?. [L2, 10M]
6. Construct the DAG for following statement. a+b*c+d+b*c [L3, 10M]
7. Construct the DAG for the following basic blocks [L3, 10M]

1. t1:=4*i
2. t2:=a[t1]
3. t3:=4*i
4. t4:=b[t3]
5. t5:=t2*t4
6. t6:=prod+t5
7. prod:=t6
8. t7:=i+1
9. i:=t7
10. if i<=20 goto 1

8. Explain the simple code generator and generate target code sequence for the following statement d:=(a-b)+(a-c)+(a-c) [L2, 10M]
9. Write short notes on i)Simple code generator [L6, 5+5M]
   ii) Register allocation

10 a) What is the role of peephole optimization in compilation process [L1, 2M]
   b) Write the issues in the design of a code generator.(any 4) [L6, 2M]
   c) Give the variety of forms in target program [L1, 2M]
   d) Give the application of DAG. [L1, 2M]
   e) Define Dead-code elimination with example. [L1, 2M]

SIDDHARTH GROUP OF INSTITUTIONS :: PUTTUR
Siddharth Nagar, Narayanavanam Road – 517583

QUESTION BANK (OBJECTIVE)

Subject with Code : Compiler Design (13A05502) **Course & Branch :** B. Tech - CSE

Year & Sem : III B.Tech & I-Sem **Regulation :** R13

UNIT-1

1. Popular type of intermediate code generation language [ ]
   A) 3 address code B) 33 address code
   C) 30 address code D) 333 address code
2. In code generation the optimizing code is converted into _____ [ ]
   A) Assembly level language B) Machine level language
   C) Mission code D) Both A & C
3. How many times the source code will be scanned is called _____ [ ]
   A) Pass B) Phase
   C) Parse D) Scanner
4. The logical operation for each part of the process of compilation is called _____[ ]
   A) Pass B) Phase
   C) Parse D) Scanner
5. Which is property of boot strapping? [ ]
   A) It must compile source language “s”
   B) It must use implementation language “i”
   C) It must generate target language “T”
   D) All the above
6. Cross compiler runs program in one program and does not produce target code for ___ [ ]
   A) Same machine B) Another machine
   C) Both A&B D) None
7. Which is not compiler construction tool _____ [ ]
   A) Parser B) Scanner generator
   C) Data flow synthesis D) None
8. Token means sequence of _____ [ ]
   A) Integers B) Floats
   C) Characters D) All the above
9. Low level programs are _____ to write [ ]
   A) Easier B) Harder
   C) Softer D) Light
10. _____ is a grouping of declarations and statements [ ]
    A) Scope B) Block
    C) Shelves D) Racks
11. ___ directly executes the operations specified in the source program on inputs supplied by the user. [ ]
    A) Interpreter B) Target program
    C) Machine language D) Assembly language
12. The _____ resolves external memory addresses. [ ]

A) Translator B) Virtual machine
C) Linker D) Pre processor

13.The _____ part breaks up the source program into constituent pieces and imposes a grammatical structure on them. [ ]
A) Synthesis B) Analysis
C) Analytical D) Syntax

14. Information about the source program and stores in _____ [ ]
A) Syntax table B) Analytical table
C) Symbol table D) Synthetic table

15. The analysis part often called_____ [ ]
A) Right end B) Left end
C) Back end D) Front end

16. The first phase of compiler is called _____ [ ]
A) Lexical analysis B) Scanning
C) Lexical scanning D) Both A&B

17. The lexical analyzer produces output in the form _____ [ ]
A) (token-name, attribute-value) B) (token-value, attribute-name)
C) (attribute-value, token-name) D) (attribute-name, token-value)

18. ___ gathers type information and saves it in either the syntax tree or the symbol table. [ ]
A) Lexical analyzer B) Syntax analyzer
C) Semantic analyzer D) Analyzer

19. An important part of semantic analyzer is _____ [ ]
A) Generating tokens B) parser tree generation
C) Type checking D) None of these

20. Syntax trees are commonly used during _____ [ ]
A) Syntax analysis B) Lexical analysis
C) Semantic analysis D) Both A & C

21. The closure of L denoted as _____ [ ]
A) L^* B) L^+
C) L* D) L^0

22. L^0 is called _____ [ ]
A) Concatenation of Zero terms B) Closure of zero terms
C) Union of Zero terms D) None

23. Which has highest precedence [ ]
A) * B) Concatenation
C) | D) All

24. Transition diagrams have collection of a nodes or circles called _____ [ ]
A) Positions B) States
C) Stages D) Edges

25. The lexical analyzer tool is called ____ [ ]
A) LUX B) LEX
C) FLEX D) LES

26. When several prefixes of the input match one or more patterns _____ [ ]
A) Always prefer longer prefix to a shorter prefix
B) Always prefer shorter prefix to a longer prefix
C) Both A&B
D) None

27. The latter file is compiled by the C compiler into a file called _____ [ ]

A) a out B) aout
C) A.Out D) a.out

28. The translation rules the form [ ]
A) Pattern {Action} B) {Pattern} Action
C) Pattern {Action} D) Pattern Action

29. The set{0,1} is the _____ [ ]
A) Decimal set B) Octal set
C) Binary set D) None

30. The _____ is a special character that cannot be a part of source program [ ]
A) Sentence B) Word
C) Sentinels D) Tokens

31. Change in one variable to change another is called _____ [ ]
A) Aliasing B) Changing
C) Differentiating D) Renaming

32. Parameters are passed from a calling procedure to the callee by _____ [ ]
A) Value B) Reference
C) Both A&B D) None

33. Analysis portion of a compiler generally separated into _____ [ ]
A) Lexical analysis & parsing B) Lexical analysis & scanning
C) Lexical analysis &syntax analysis D) Lexical analysis &semantic analysis

34. A _____ is a description of form that the lexeme of a token may take [ ]
A) Syntax B) Procedure
C) Pattern D) Function

35. _____ is any finite set of symbols [ ]
A) Strings B) Characters
C) Alphabets D) Numbers

36. _____ is example of alphabet used in software systems. [ ]
A) C B) D
C) ASCII D) Both A & B

37. Finite sequence of symbols is called _____ [ ]
A) String B) Words
C) Sentence D) All the above

38. If x=dog y=house then xy= ____ [ ]
A) Dog House B) doghouse
C) DOGHOUSE D) DogHouse

39. If x & y are strings, then the concatenation of x & y denoted _____ [ ]
A) x*y B) xy
C) XY D) X*Y

40. The Positive closure of L denoted as _____ [ ]
A) L^* B)L^+
C) L* D) L^0

UNIT-2

1. Context Free grammar production rule ____ [ ]
   A) a->b B) A->b
   C) A->B D) A->α
2. S->AB,A->a/b, B->b grammar can produce ____ strings [ ]
   A) b B) ba
   C) AB D) None
3. Which one is not a LR(0) item ____ [ ]
   A) A->. Xyz B) A->x.yz
   C) A->xyz D) None
4. FIRST (a) ____ [ ]
   A) A B) a
   C) Both A & B D) None
5. In LL(1) first L stands for ____ [ ]
   A) Left most derivation B) Scanning from left to right
   C) Both A & B D) None
6. The parsing table has no multiple entries is set to be ____ [ ]
   A) LL(1) B) LL(0)
   C) Predictive parser D) Non recursive parser
7. Follow(start symbol of grammar) should add ____ [ ]
   A) + B) a
   C) A D) $
8. In shift action the input symbol is ____ [ ]
   A) Shifted to the stack B) Reduced with non terminal
   C) Both A & B D) None
9. In synthesized attribute node value is calculated from ____ [ ]
   A) Leaves to root B) From top to bottom
   C) Both A & B D) None
10. In Lex specifications the translation rules starts and ends with ____ [ ]
    A) %,% B) Begin, End
    C) %%,%% D) Start ,Stop
11. A parser which is a variant of top-down parsing without backtracking is_____ [ ]
    A) Recursive Descend B) Operator Precedence
    C) LL (1) parser D) LALR Parser
12. The legal text which is derived from a distinguished symbol is called ____ [ ]
    A) Axioms B) lexemes
    C) sentence symbol D) both A & C
13. A LL parser is also known as [ ]
    A) Top down parser B) Bottom up parser
    C) LL(0)parser D) LL(1)parser
14. Symbols that cannot be replaced are known as _____ [ ]
    A) Non-terminals B) Terminals
    C) Symbols D) tokens
15. Terminals represent character strings that are recognized by _____ [ ]
    A) Syntax analyser B) lexical analyser
    C) semantic analyzer D) none of these
16. What the letter ‘T’ represents in production symbols ____ [ ]
    A) Term B) Token

C) Table D) None of these

17. Yaac is available as a command on _____ [ ]
A) MNIX B) UNIX
C) DOS D) None of these

18. The process which starts from the leaf node and ends with the starting symbol is known as _____ [ ]
A) Top down parsing B) Bottom up parsing
C) Recursive parsing D) LL(1)parser

19. Which action in the shift reduce parsing detect the syntax errors [ ]
A) ACTION B) GOTO
C) Error D) Reduce

20. The simplest method for shift reduced parser is known as _____ [ ]
A) SLR B) LALR
C) CLR D) LR

21. _____ is an attribute whose value at a node in a parse tree is defined in terms of attribute at the parent and/or sibling of that node. [ ]
A) L-attribute B) S-attribute
C) Synthesized D) Inherited

22. In shift action the input symbol is ____ [ ]
A) Shifted to the stack B) Reduced with non terminal
C) Both A & B D) None

23. _____ is a recursive descent parser that needs no backtracking. [ ]
A) Predictive Parser B) LR
C) Brute Force D) Shift Reduce

24. An attribute grammar in which all attributes are__ then it is called S attributed grammar. [ ]
A) Parsed B) Inherited
C) A-attributed D) synthesized

25. The Output From second phase ____ [ ]
A) Parse tree B) Intermediate Code
C) Tokens D) None

26. The Output From last phase ____ [ ]
A) Parse tree B) Syntax tree
C) Assembly language D) Both A & B

27. In parse tree the leaf node is labeled by ____ [ ]
A) Epsilon (€) B) Terminal
C) Non terminal D) Start symbol of grammar

28. In top down parsers the parse tree constructed from ____ [ ]
A) Bottom to top B) Top to bottom
C) Both A & B D) None

29. E-> E*E consists ____ [ ]
A) Left factoring B) Left recursion
C) Both A & B D) None

30. In LL(1) first L stands for ____ [ ]
A) Left most derivation B) Scanning from left to right
C) Both A & B D) None

31. FIRST(+) ____ [ ]
A) + B) +,-
C)+,-,* D)None

32. The parsing table has no multiple entries is set to be ____ [ ]
A) LL(1) B) LL(0)
C) Predictive parser D) Non recursive parser

33. Which of the following is most powerful bottom up parser? [ ]
A) SLR B) LALR
C) CLR D) Operator Precedence

34. In Reduce action the input symbol is ____ [ ]
A) Shifted to the stack B) Reduced with non terminal
C) Both A & B D) None

35. LALR(1) grammar is ____ [ ]
A) LR(0) B) SLR(1)
C) LR(1) D) None

36. In bottom up parsing string is generated from ____ [ ]
A) RMD in reverse order B) Leftmost derivation
C) Both A & B D) None

37. A parser which is a variant of top-down parsing without backtracking is_____ [ ]
A) Recursive Descend B) Operator Precedence
C) LL(1) Parser D) LALR Parser

38. The legal text which is derived from a distinguished symbol is called____ [ ]
A) Axioms B) Lexemes
C) Sentence symbol D) Both A & C

39. A LL parser is also known as [ ]
A) Top down parser B) Bottom up parser
C) LL(0)parser D) LL(1)parser

40. The end of file is represented by the special symbol is ____ [ ]
A) $ B) υ
C) μ D) π

UNIT-3

1.Advantage of panic mode of error recovery is that_____ [ ]

A)It is simple to implement B)It never gets into an infinite loop

C)Both A) and B) D) None of these

2.An Intermediate code form is_____ [ ]

A)Postfix notation B)Syntax trees

C)Three address code D)All of these

3.Intermediate code generation phase gets input from_____ [ ]

A)Lexical analyser B) Syntax analyser

C)Semantic analyser D)Error Handling

4.Relocating bits used by relocating loader are specified by_____ [ ]

A)Relocating loader itself B)Linker

C)Assembler D)Macro Processor

5.Reduction in strength means [ ]

A)Replacing run time computation by compile time computation

B)Removing loop invariant computation

C)Removing common sub expression

D)None of these

6. The computer languages are generally translated into _____ [ ]

A) Assembly B) Machine

C) Pascal D) FORTRAN

7.Any statement that immediately follows a goto or conditional goto statement in a sequence of three address statements is a _____. [ ]

A) Leader B) Instructor

C) A& B D) none

8.General Form of a three-address statement is _____ [ ]

A)a:=b (op) c B) a:=b c

C) a:=b D)None of these

9. The value of _____ attribute is computed from the value of attributes at the siblings and parent of that node. [ ]

A) S-attribute B) Synthesized

C) Inherited D) All Above

10 Synthesized attributes can be easily simulated by _____ [ ]

A)LL grammar B)LR Grammar

C) Operator grammar D) Ambiguous grammar.

11.A Parse tree showing the values of attributes at each node is called_____ [ ]

A) Syntax B) Augmented

C) Annotated D) Semantic

12. S-attribute definition is also called as_____. [ ]

A)Postfix SDT B) Prefix SDT

C) SDT D)none

13. The information associated with data variables are_____ [ ]

A)Name B)Data types
C)Scopes, binding& life time D)All Above

14._____ are the examples for high level Intermediate languages. [ ]
A)Abstract Syntax Tree (AST) B)Postfix notation
C) both A & B D) none of these

15. Three address code is a _____level intermediate language [ ]
A) High B) Medium
C) Low D) none

16._____ is the process of replacing a function call with the body of the called function. [ ]
A) Inlining function B) friend function
C) both A & B D) none

17.The intermediate optimized code can be sequence of _____ [ ]
A) Quadruples B)target code
C)source program D)binary language

18. Variable descriptors are also known as ___ [ ]
A) register descriptor B)address variables
C) pseudo registers D)constants

19.An _____ attribute at node N is defined only in terms of attribute values at N's parent, N itself and N's siblings. [ ]
A) synthesized B) Inherited
C) A & B D) None

20.A _____ graph depicts the flow of information among the attribute instances in a particular parse tree. [ ]
A) dependency B) annotated parse
C) syntax D) none

21.The fields of Triples are op,arg1 & ____ [ ]
A) Result B) arg2
C) operands D) none

22.object program is a _____ [ ]
A) Program written in machine language
B) Program to be translated into machine language.
C) Translation of high level language into machine language.
D) none of these

23.When is the type checking usually done? [ ]
A) During syntax directed translation B) During lexical analysis
C) During code optimization D) During syntax analysis

24. Implicit type conversions, is_____ [ ]
A)Done automatically by the compiler. B) Done automatically by the User
C) Done automatically by the OS D) None of these

25.In synthesized attribute node value is calculated from ____ [ ]
A) Leaves to root B)From top to bottom
C)Both A & B D)None

26. _____is an attribute whose value at a node in a parse tree is defined in terms of attribute at the parent and/or sibling of that node. [ ]

A) L-attribute B)S-attribute

C) Synthesized D) Inherited

27. An attribute grammar in which all attributes are_____ then it is called S attributed grammar. [ ]

A) Parsed B)Inherited

C) A-attributed D) synthesized

28. An attribute grammar in which all the attributes are synthesized is called____Attributed grammar. [ ]

A) P B)Q

C) R D) S

29. A _____ keeps the information about each register. [ ]

A) Register descriptor B) address descriptor

C) variable descriptor D) none of these

30. In quadruple notation maximum_____ fields are used to represent operands. [ ]

A) 1 B)2

C) 3 D) 4

31. Which of the following is an intermediate code form_____ [ ]

A) Three address code B) syntax tree

C) parser D) none of these.

32.A parse tree showing the values of its attribute is called _____ [ ]

A)Dependency Graph B)parse tree

C) Annonated parse tree D) None

33. A Type name is_____ [ ]

A)Type expression B)Type Checking

C) Backpatching D) None

34. Makelist(i) is a function of_____ [ ]

A)Type expression B)Type Checking

C) Backpatching D) None

35. Syntax directed translation scheme is desirable because ___ [ ]

A)It is based on the syntax

B)It is easy to modify

C)Its description is independent of any implementation

D)All of these

36. If Conversion from one type to another type is done automatically by the complier then, it is called__ [ ]

A)Implicit conversion B)Coercions

C) Both A & B D) None of these

37.The term environment in programming language semantics is said as ____ [ ]

A) Function that maps a name to value held there

B)Function that maps a name to storage location

C)The function that maps a storage location to the value held there

D)None of the above

38.A self relocating program is one which _____ [ ]

A) cannot be made to execute in any area of storage other than the designated for it at the time of its coding or translation

B) Consists of a program and relevant information for its relocation

C) Can itself perform the relocation of its address sensitive protions

D) All of the above

39.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) Be evaluated only if the definition has synthesized attributes

D) None of the above

40. Generation of intermediate code based on a abstract machine model is useful in compilers because [ ]

A) it makes implementation of lexical analysis and syntax analysis easier

B)syntax directed translation can be written for intermediate code generation.

C)It enhances the portability of the front end of the compiler

D)it is not possible to generate code for real machines directly from high level language programs

UNIT-4

1. The symbol table Implementation is based on the property of locality of reference is ___ [ ]

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

2.Which one of the following is an object code form? [ ]

A)Absolute machine language B) Re-locatable machine code
C) Assembly language D) all of the above

3.The statement of the form a:=b is called a _____ statement. [ ]

A)Common B) Copy
C) Induction Variable D) Decode

4.To check whether a variable is exactly defined once or not is a _____ check. [ ]

A) Uniqueness check B) Flow of Control Check
C) name check D) Above all

5. ___ is a Data Structure, which is used by compiler to keep track of information [ ]

A) Lexical analyser B) Symbol Table
C) Semantic Table D) Semantic Analyzer

6. A symbol is said to be ____if it has different meaning depending on its context or use. [ ]

A) Override B)Overloaded
C) Overwrite D) None

7.The storage strategy in which activation record is maintained even after the execution of a procedure is completed. [ ]

A) Stack Allocation B) Heap Allocation
C) Static Allocation D) Dynamic Allocation

8.An optimized compiler can perform _____ [ ]

A) Optimize the code B) Expand the Code
C) Reporting Errors D) Above all.

9.Machine independent optimization is_____ [ ]

A) Register Allocation B) Frequency reduction
C) Data intermixed with instruction D) None

10. A _____ can be visualized as a set of records in data structure. [ ]

A)Symbol table B) Variable
C) both A & B D) none

11.A hash function should produce the same hash value for two different keys then it is called _____ [ ]

A)collisions B) Heap allocation
C) Stack allocation D) none of these.

12. Set of information constitute a record in dynamic allocation is called _____ [ ]

A)Activation Record (AR) B)frame.
C) both A & B D) none of these

13. The process of replacing the costly instruction by cheaper one is called ____. [ ]

A)Strength Reduction B)Strength Increases
C) Induction Variable D) none

14.A movement of data from memory to register or register to memory is considered as _____ cost. [ ]
A) code B) unit
C) register D) none

15.SISD full form _____ [ ]
A)Single Instruction Single Data B)Set of Instruction Single Data
C) Single Instruction Set of Data D) none of these.

16.The information associated with operations is _____ [ ]
A) Operators B)Functions ,Function arguments
C)Scope and visibility D) all above

17. The _____ rules of a language determine which declaration of the name applies when the name appears in the text of a program. [ ]
A) Life time B)Alias
C) Scope D) none

18.A technique for improving the quality of a target code locally by examining a short sequence of target instructions and replacing with faster sequence is called _____optimization. [ ]
A) Peephole B) Procedural
C) Flow graph D) none

19.Which one of the following is a symbol table attribute? [ ]
A) scope B)Data types
C) names D)all above

20.Replacing multiplication operator with addition operator is _____ [ ]
A) Constant Folding B)Reduction in Strength
C) Copy Propagation D)None

21.A Transformation is called ____ if it can be performed in single basic block [ ]
A)Local Optimization B)Global optimization
C)Both A&B D)None

22. The transformations that are applied on the multiple basic blocks is called as_[ ]
A) Global Optimization B) Local Optimization
C) Block Optimization D) none

23. Which is not an example for function-preserving transformation? [ ]
A)Copy propagation B)Flow of control
C)Constant folding D)dead-code elimination

24. Optimization is_____ phase in compilation process [ ]
A)1 B)2
C)5 D)4

25. A:=B+C find use and definition variables [ ]
A)use=A,B ,Def=C B) use=A ,Def= B,C
C) use=B,C ,Def=A D) None

26. If x:=y is a statement then copy propagation is a kind of transformation in which we use [ ]

A)x for y B)y for x

C)x=y D)none

27.If the variable contain its value and used subsequently then variable is said to be __ [ ]

A)Live B)Dead

C)Alive D)All

28. Certain code moving outside of the loop is_____ [ ]

A)code notion B)induction variable

C)code motion D)strength reduction

29. Copy statement is ___ [ ]

A)X=a+b B)a[i]=x

C)x++ D)None

30. Busy expressions are useful in performing _____ optimization [ ]

A)local B)global

C)loop D)code movement

31.The process of eliminating the repeated statements in intermediate code [ ]

A)Code optimization B)Code generation

C)Efficiency D)implementation

32. One of the purposes of using intermediate code in compilers is to [ ]

A) make parsing and semantic analysis simpler.

B) improve error recovery and error reporting

C) increase the chances of reusing the machine-independent code optimizer in other compliers.

D) improve the register allocation

33. Which of the following statements are CORRECT? [ ]

1) Static allocation of all data areas by a compiler makes it impossible to implement recursion.

2) Automatic garbage collection is essential to implement recursion.

3) Dynamic allocation of activation records is essential to implement recursion.

4) Both heap and stack are essential to implement recursion.

A) 1 and 2 only B) 2 and 3 only

C) 3 and 4 only D) 1 and 3 only

34. Which one of the following is not an addressing mode? [ ]

A) Register indirect B)Auto increment

C)Relative indexed D)Immediate operand

35. Computers can have instruction formats with___ [ ]

A)Only two address and three address instructions

B)Only one address and two address instructions

C)Only one address, two addresses and three address instructions

D) Zero address, one address, two addresses and three address instructions

36. The identification of common sub expression and replacement of run time computation by compile- time computation is [ ]

A)loop optimization B) local optimization

C) constant folding D) data flow analysis

37. The graph that shows basic blocks and their successor relationship is called [ ]

A) DAG B) Flow graph

C) Control graph D) Hamiltonion graph

38.The specific task storage manager performs [ ]

A) allocation/deallocation

B) Protection of storage area allocated to a program from illegal access by other programs in the system

C) the status of each program

D) both A & B

39. Pick the machine independent phase of the compiler [ ]

A)Syntax analysis B) Lexical analysis

C) Intermediate code generation D) all of the above

40.Type checking is normally done during [ ]

A) Lexical Analysis B) Syntax Analysis

C) Syntax Directed Translation D) Code generation

UNIT-5

1.Determining Common sub expression can be done using____ [ ]
A)Compiler B) Interpreter
C)DAG D) parse tree

2. In DAG the interior nodes are labeled with _____ [ ]
A) Number in BFS B) Special colors
C) Identifier D) Number in BFS

3.A_____ occurs when there is a reference to storage that has been de-allocated. [ ]
A)Dangling reference B) if-else
C) Register allocation D) none

4. The process of moving the statement from one part of the program to another is called _____ [ ]
A) Code motion B) Constant folding
C) Copy propagation D) none

5. _____ is a simple, systematic technique for allocating registers and managing register spills. [ ]
A) DAG B) Graph coloring
C) A & B D) none

6. The _____ is a node in the flow graph, which precedes all the statements in the loop. [ ]
A) Numerator B)Dominator
C) tree D)none

7. Output of code generator is_____ [ ]
A)Source code B)Intermediate code
C)Assembly code D)None of these

8. At a point in a program if the value of the variable can be used subsequently, then that variable is _____ Variable. [ ]
A)Live B)Dead
C)Loop D)None of these

9.The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called _ [ ]
A) Assembly B) Parsing
C) Relocation D) Symbol resolution

10.The runtime environment of the language must be able to collect the unused memory automatically for avoiding the memory leak, which is called _____ [ ]
A) Garbage collection B) Memory allocation
C) Optimization D) none

11._____decides whether the variable is alive at that point [ ]
A)Reaching definitions B)Available Expressions
C)Live Variables D) None

12. In Reachable Definition the output of the block is calculated by Output[B] = _[ ]
A)Spawn[B] U (Input[B] – Delete[B]) B) Spawn[B] U Input[B]
C) Spawn[B] UDelete[B] D) none of these.

13.The transformations that are applied across the basic blocks is called as____ [ ]
A) Global Optimization B) Local Optimization
C) Block Optimization D) none

14. __is the portion of the program which will not be executed in any path of the program.

[ ]

A) Live code B) Dead Code
C) reachable Code D) none of these

15Acronym for DAG ____ [ ]

A) Directed Acyclic Graph B)Direct Cyclic Graph
C)Derived Acyclic Graph D)Deviated Acyclic Graph

16.In DAG the interior nodes are labeled by _____ symbol [ ]

A) Operands B) operator
C) both A & B D) none of these.

17. The first statement in basic block is_____ [ ]

A) Main Statement B)Follow
C)Header D)Leader

18.Running time of a program depends on _____ [ ]

A)The way the register and addressing modes are used
B)The Order in which computations are performed
C)The usage of machine idioms
D) All of these

19.Which of the following does not interrupt a running process? [ ]

A)A device B)Timer
C)Scheduler D)Power failure

20.The optimization technique which is typically applied on loops is__ [ ]

A) Peephole optimization B) Removal of invariant computation
C)Constant folding D)All of these

21.The optimization which avoids test at every iteration is____ [ ]

A)Loop Unrolling B)Loop jamming
C)Constant folding D)None of these

22. We can optimize code by___ [ ]

A)Dead code elimination B)Common Subprograms
C)Copy intermediate loop D)Loop Declaration

23.Input to code generator is_____ [ ]

A)Source code B)Intermediate code
C)Target code D)All of the above

24.Local and loop optimization in turn provide motivation for_____ [ ]

A)Data flow analysis B)Constant folding
C)Peephole optimization D)DFA and Constant folding

25. A Symbol table is _____ [ ]

A)Data structure B)Variable
C)Data Type D)None

26.Live variables are used in_____ elimination [ ]

A)Common sub Expression B)Copy Propagation
C)Code Motion D)Dead code

27.DAG is constructed from _____ [ ]

A)3 address code B)program
C)blocks D)none

28.An estimate of how frequently a variable used is ___ [ ]

A)Usage count B)Reference count
C)Program count D)Process count

29.A flow graph is a directed graph in which the flow control information is added to the _____ [ ]

A)blocks B)graph
C)tree D)basic blocks

30. Code generation phase converts the __into a sequence of machine instruction [ ]
A) Intermediate optimized code B) assembly code
C) target code D) none of these

31. Peep-hole Optimization is a form of [ ]
A) Local Optimization B)Constant Folding
C)Copy Propagation D)None of these

32. If the value of the variable is changed every time then that variable is called as [ ]
A)invarient variable B)Dead variable
C)Live variable D)Induction variable

33. Basic block is Sequence of _____ [ ]
A)Statements B)Loops
C)Values D)None

34 _____technique performed on target code [ ]
A) Local Optimization B)Loop optimization
C)Peep-hole Optimization D)None

35. The strength reduction is related to _____ [ ]
A)variables B)Loops
C)operators D)All

36. The local optimization performed its scope in certain specific block is known as ____ [ ]
A)Local scope B)Global scope
C)Dynamic scope D)static scope

37.Graph coloring is strategie of _____ [ ]
A)Register allocation B)Heap allocation
C) Stack allocation D)None

38. _____ keeps frequently used value in a fixed register throughout a loop. [ ]
A) Local register allocation B) global register allocation
C) static allocation D) none of these

39. DAG representation of a basic block allows [ ]
A) Automatic detection of local common sub expressions
B) Automatic detection of induction variables
C) Automatic detection of loop variant
D) None of the above

40. Local and loop optimization in turn provide motivation for [ ]
A) Data flow analysis B) Constant folding
C) Pee hole optimization D) DFA and constant folding