SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai - 600089

**FACULTY OF ENGINEERING AND TECHNOLOGY**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

****

**QUESTION BANK**

**DEGREE / BRANCH: B.TECH-CSE & CSE WITH SPECIALIZATION**

**VI SEMESTER**

**18CSC304JT – COMPILER DESIGN**

**2018 Regulation**

**Academic Year 2022-2023 EVEN SEMESTER**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

**Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai-600089**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**QUESTION BANK**

**SUBJECT : 18CSC304J – COMPILER DESIGN**

**SEM/ YEAR: III/V**

**Course Outcomes**

CO-1 : Acquire the knowledge of mathematics and engineering principles for the Design of Compilers

CO-2 : Acquire the ability to identify specification of a language's lexical rules of Lexical Analyzer

CO-3 : Apply the knowledge of Syntax Analyzer for parsing the sentences in a compiler grammar

CO-4 : Understand the concepts of translation of various intermediate codes .

CO-5 : Apply the knowledge to implement Code Generator for compilers

CO-6 : Analyze and Design the methods of developing a Code Optimizer

| **UNIT I** | | | |
| --- | --- | --- | --- |
| Compilers – Analysis of the source program-Phases of a compiler – Cousins of the Compiler-Grouping of Phases – Compiler construction tools- Lexical Analysis – Role of Lexical Analyzer-Input Buffering- Specification of Tokens--Finite automation – deterministic Finite automation - non deterministic-Transition Tables- Acceptance of Input Strings by Automata-State Diagrams and Regular Expressions- Conversion of regular expression to NFA - Thompson’s method-Conversion of NFA to DFA- Simulation of an NFA-Converting Regular expression directly to DFA- Minimization of DFA-Minimization of NFA- Design of lexical analysis (LEX) | | | |
| **PART-A (Multiple Choice Questions)** | | | |
| **Q.**  **No** | **Questions** | **Course Outcome** | **Competence**  **BT Level** |
| **1** | **A Compiler is**  a)A program that place program into memory and prepares them for execution  b)A program that automates the translation of assembly language into machine language  **c)A program that accepts program written in high level language and produces an object program**  d)A program that appears to execute a source program as if it were machine language | CO1 | BT 1 |
| **2** | **Compiler can check \_\_\_\_\_\_\_\_ error.**  a) Logical  **b) Syntax**  c) Content  d) Both A and B | CO1 | BT 1 |
| **3** | **Grammar is checked by which component of compiler**  a) Scanner  **b) Parser**  c) Semantic Analyzer  d) None of the mentioned | CO1 | BT 1 |
| **4** | **In a compiler, the data structure responsible for the management of information about variables and their attributes is**  a) Semantic stack  b) Parser table  **c) Symbol table**  d)Syntax tree | CO1 | BT 1 |
| **5** | **How many tokens are there in following statement ?Printf(“j=%d,&amp;j=%x”,j&amp;j)**  a)4  b) 5  c) 9  **d) 10** | CO2 | BT 1 |
| **6** | **When is the type checking usually done?**  **a)Syntax directed translation**  b)lexical analysis  c)code optimization  d) syntax analysis | CO1 | BT 1 |
| **7** | **How many DFA’s exits with two states over input alphabet {0,1} ?**  a) 16  b) 26  c) 32  **d) 64** | CO2 | BT 2 |
| **8** | **Regular expression for all strings starts with ab and ends with bba is.**  a) aba\*b\*bba  b) ab(ab)\*bba  **c) ab(a+b)\*bba**  d) All of the mentioned | CO2 | BT 1 |
| **9** | **Transition function maps.**  a) Σ \* Q -> Σ  b) Q \* Q -> Σ  c) Σ \* Σ -> Q  **d) Q \* Σ -> Q** | CO1 | BT 2 |
| **10** | **Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least.**  a) N^2  **b) 2^N**  c) 2N  d) N! | CO2 | BT 2 |
| **11** | **In a compiler, keywords of a language are recognized during**  a)parsing of the program  b)the code generation  c)**the lexical analysis of the program**  d)dataflow analysis | CO1 | BT 2 |
| **12** | **Compiler should report the presence of \_\_\_\_\_\_\_\_\_\_ in the source program, in translation process.**  a) Classes  b) Objects  **c) Errors**  d) Text | CO2 | BT 1 |
| **13** | **What is the output of lexical analyzer?**  a) A parse tree  b) **A list of tokens**  c) Intermediate code  d) Machine code | CO2 | BT 1 |
| **14** | **An NFA’s transition function returns**  a) A Boolean value  b) A state  **c) A set of states**  d) An edge | CO2 | BT 2 |
| **15** | **Which of the following is used for grouping of characters into tokens (in a computer)**  a)A parser  b) Code optimizer  c) Code generator  **d)Scanner** | CO2 | BT 1 |
| **16** | **Evaluate the number of states require to accept string ends with 10.**  **a) 3**  b) 2  c) 1  d) can’t be represented. | CO1 | BT 2 |
| **17** | Which of the following is application of finite automata?  a)Compiler design  b)Grammar Parsers  c)Text Search  **d) All the above** | CO2 | BT 1 |
| **18** | **Which of the following are language processor?**  **a) Assembler**  b) Compiler  c) Interpreter  d) All the above | CO1 | BT 1 |
| **19** | In regular expressions, the operator ‘\*’ stands for?   1. Concatenation 2. Selection’ 3. **Iteration** 4. Addition | CO2 | BT 1 |
| **20** | What is the relation between NFA accepted languages and DFA accepted languages?   1. > 2. < 3. **=** 4. **<=** | CO2 | BT 1 |
| **PART B (4 Marks)** | | | |
| 1 | Interpret the functions of lexical analyzer? | CO2 | BT 2 |
| 2 | List the phases that constitute the front end of a compiler | CO1 | BT 1 |
| 3 | What is a symbol table? | CO1 | BT 1 |
| 4 | Justify, how will you group the phases of compiler? | CO1 | BT 1 |
| 5 | Compare and Contrast between compiler and interpreter? | CO1 | BT 2 |
| 6 | Manipulate the transition diagram for an identifier. | CO1 | BT 1 |
| 7 | Summarize the algebraic properties of regular expressions? | CO1 | BT 1 |
| 8 | Compare the features of DFA and NFA. | CO2 | BT 2 |
| 9 | What is a finite automaton? | CO2 | BT 2 |
| 10 | Classify the possible error recovery actions in lexical analyzer? | CO2 | BT 2 |
| 11 | What are the issues of the lexical analyzer? | CO2 | BT 2 |
| 12 | Indicate the role of lexical analyzer? | CO2 | BT 2 |
| 13 | Write short notes on input buffering. | CO1 | BT 2 |
| 14 | Mention the back-end phases of a compiler | CO1 | BT 2 |
| 15 | Order the functions performed in analysis phase? | CO1 | BT 2 |
| 16 | Order the functions performed in synthesis phase? | CO1 | BT 2 |
| 17 | What are the factors affecting number of passes in compiler? | CO1 | BT 2 |
| 18 | What are compiler construction tools? | CO1 | BT 2 |
| 19 | Mention few cousins of compiler | CO1 | BT 2 |
| 20 | Define token, pattern and lexeme. | CO2 | BT 1 |
| **PART C (12 Marks)** | | | |
| **1** | Define Compiler & List out its Phases. | CO 1 | BT 1 |
| **2** | 1. Discuss the role of lexical analyzer in detail. 2. Explain briefly the design of lexical analyzer | CO 1 | BT 1 |
| **3** | Give the minimized DFA for the following expression. (a/b)\*abb using direct method | CO 2 | BT 3 |
| **4** | 1. Explain the compiler construction tools 2. Explain specification and recognition of tokens | CO 1 | BT 2 |
| **5** | Give the minimized DFA for the following expression | CO 2 | BT 3 |
| **6** | Explain input buffering in detail and Write briefly on the issues in input buffering and how they are handled | CO 2 | BT 2 |
| **7** | Give the minimized DFA for the following expression. (a/b)\*abb using tabulation method | CO 2 | BT 3 |
| **8** | Prove the following two regular expressions are equivalent by showing the minimum state DFA’s are same.  i) (a/b)\* ii) (a\*/b\*) | CO 2 | BT 3 |
| **9** | Explain the design of lexical analyzer using LEX | CO 2 | BT 2 |

**Note:**

1. **BT Level –** Blooms Taxonomy Level
2. **CO – Course Outcomes**

BT1 – Remember BT2 – Understand BT3 – Apply BT4 – Analyze BT5 – Evaluate BT6 – Create