1. Which phase of the compiler is responsible for generating intermediate code?

a) Syntax Analysis
b) Semantic Analysis
c) Intermediate Code Generation
d) Code Optimization

Answer: c) Intermediate Code Generation

Explanation: Intermediate code generation is the phase in the compiler where high-level code is transformed into a simpler intermediate representation suitable for further processing.

2. Which of the following is NOT a typical construct handled during intermediate code generation?

a) Declarations
b) Loops
c) Boolean expressions
d) Case statements

Answer: b) Loops

Explanation: While loops, for loops, and other loop constructs are usually handled during high-level semantic analysis and later during code generation and optimization phases.

3. What is the purpose of back patching in intermediate code generation?

a) To optimize generated code

b) To resolve addresses in jump statements
c) To perform type checking
d) To handle variable declarations

Answer: b) To resolve addresses in jump statements

Explanation: Back patching is a technique used to fill in the target addresses of jump statements (like goto or if statements) once the target is known.

4. Which phase of the compiler is responsible for allocating registers and assigning them to variables?

a) Lexical Analysis
b) Intermediate Code Generation
c) Code Optimization
d) Register Allocation

Answer: d) Register Allocation

Explanation: Register allocation is a phase in code generation where physical or virtual registers are assigned to variables to minimize memory accesses.

5. In compiler design, what is a basic block?

a) A block of code without any control flow statements
b) A block of code with only one entry point and one exit point
c) A block of code representing a loop
d) A block of code with multiple exit points

Answer: b) A block of code with only one entry point and one exit point

Explanation: A basic block is a sequence of code with one entry point and one exit point; control flows into the block at the entry point and flows out at the exit point.

6. Which data structure is commonly used to represent the control flow of a program during code generation?

a) Linked List
b) Tree
c) Graph
d) Stack

Answer: c) Graph

Explanation: Control flow graphs (CFGs), which represent the flow of control within a program, are commonly represented using graph data structures.

7. What is the primary goal of peephole optimization?

a) To optimize the use of registers
b) To optimize the size of the generated code
c) To optimize the control flow
d) To optimize memory access

Answer: b) To optimize the size of the generated code

Explanation: Peephole optimization is a technique where the compiler looks for and applies optimizations within small, localized sections of generated code, primarily aiming to reduce code size.

8. Which phase of the compiler typically performs peephole optimization?

a) Lexical Analysis
b) Intermediate Code Generation
c) Code Optimization
d) Code Generation

Answer: c) Code Optimization

Explanation: Peephole optimization is usually performed during the code optimization phase, where the compiler analyzes generated code and applies various optimizations.

9. What is the purpose of a DAG representation of basic blocks?

a) To simplify complex control flow
b) To optimize memory access
c) To represent code in a graphical format
d) To identify common subexpressions

Answer: d) To identify common subexpressions

Explanation: Directed Acyclic Graphs (DAGs) are often used to identify and eliminate redundant computations by representing common subexpressions in a compact form.

10. Which phase of the compiler involves converting DAGs back into executable code?

a) Lexical Analysis
b) Intermediate Code Generation
c) Code Optimization
d) Code Generation

Answer: d) Code Generation

Explanation: After optimization, DAGs are typically translated back into executable code during the code generation phase of the compiler.

11. Which of the following is NOT an issue in the design of a code generator?

a) Register allocation
b) Memory management
c) Optimization techniques
d) Lexical analysis

Answer: d) Lexical analysis

Explanation: Lexical analysis deals with tokenizing source code, which is not directly related to the design of a code generator.

12. Which of the following is a primary concern in register allocation and assignment?

a) Minimizing compilation time
b) Reducing the size of the source code
c) Minimizing the number of register spills
d) Maximizing the number of temporary variables

Answer: c) Minimizing the number of register spills

Explanation: Register allocation aims to minimize the need for storing variables in memory (register spills), which can degrade performance.

13. Which optimization technique focuses on identifying and eliminating redundant computations?

a) Loop optimization
b) Peephole optimization
c) Common subexpression elimination
d) Inline expansion

Answer: c) Common subexpression elimination

Explanation: Common subexpression elimination is a technique that identifies repeated computations and replaces them with a single computation, thus optimizing the generated code.

14. What is the purpose of basic blocks in code generation?

a) To group related instructions together
b) To represent loops and conditional statements
c) To simplify the control flow of the program
d) To allocate memory for variables

Answer: a) To group related instructions together

Explanation: Basic blocks are used to organize instructions into cohesive units, simplifying analysis and optimization of code.

15. Which phase of the compiler focuses on rearranging code to improve performance without changing its semantics?

a) Lexical Analysis
b) Code Optimization
c) Intermediate Code Generation
d) Syntax Analysis

Answer: b) Code Optimization

Explanation: Code optimization aims to improve the efficiency of generated code by applying various transformations while preserving the program’s behavior.

16. In a flow graph, what do nodes represent?

a) Instructions
b) Variables
c) Expressions
d) Control flow paths

Answer: d) Control flow paths

Explanation: In a flow graph, nodes represent points in the program where control flow decisions are made, such as branches, loops, or function calls.

17. Which of the following is NOT typically considered in register allocation?

a) Minimizing register spills
b) Minimizing memory usage
c) Maximizing the number of available registers
d) Minimizing the number of live ranges

Answer: b) Minimizing memory usage

Explanation: Register allocation focuses on minimizing the need for accessing memory by maximizing the use of registers for variables.

18. What is the main advantage of using a DAG representation in code generation?

a) Reducing compilation time
b) Simplifying debugging
c) Identifying common subexpressions
d) Improving code readability

Answer: c) Identifying common subexpressions

Explanation: DAG representation helps in identifying common subexpressions, which can be eliminated to improve the efficiency of generated code.

19. Which optimization technique involves replacing a function call with the actual body of the function?

a) Function inlining

b) Loop unrolling
c) Dead code elimination
d) Constant folding

Answer: a) Function inlining

Explanation:

Function inlining replaces a function call with the actual body of the function, potentially improving performance by eliminating the overhead of function calls.

20. What is the primary goal of generating code from a DAG representation?

a) To minimize compilation time
b) To improve code readability
c) To optimize memory access
d) To efficiently represent common subexpressions

Answer: c) To optimize memory access

Explanation: Generating code from a DAG representation allows for efficient memory access by identifying and eliminating redundant computations and memory accesses.

Related posts:

1. Introduction to compiling & Lexical Analysis MCQs
2. Syntax Analysis &Syntax Directed Translation MCQs
3. Type Checking & Run Time Environment MCQs

4. Code Optimization MCQs
5. Introduction to Energy Science MCQ
6. Ecosystems MCQ
7. Biodiversity and its conservation MCQ
8. Environmental Pollution mcq
9. Social Issues and the Environment MCQ
10. Field work mcq
11. Discrete Structure MCQ
12. Set Theory, Relation, and Function MCQ
13. Propositional Logic and Finite State Machines MCQ
14. Graph Theory and Combinatorics MCQ
15. Relational algebra,Functions and graph theory MCQ
16. Data Structure MCQ
17. Stacks MCQ
18. TREE MCQ
19. Graphs MCQ
20. Sorting MCQ
21. Digital Systems MCQ
22. Combinational Logic MCQ
23. Sequential logic MCQ
24. Analog/Digital Conversion, Logic Gates, Multivibrators, and IC 555 MCQ
25. Introduction to Digital Communication MCQ
26. Introduction to Object Oriented Thinking & Object Oriented Programming MCQ
27. Encapsulation and Data Abstraction MCQ
28. MCQ
29. Relationships – Inheritance MCQ
30. Polymorphism MCQ

31. Library Management System MCQ
32. Numerical Methods MCQ
33. Transform Calculus MCQ
34. Concept of Probability MCQ
35. Algorithms, Designing MCQ
36. Study of Greedy strategy MCQ
37. Concept of dynamic programming MCQ
38. Algorithmic Problem MCQ
39. Trees, Graphs, and NP-Completeness MCQ
40. The Software Product and Software Process MCQ
41. Software Design MCQ
42. Software Analysis and Testing MCQ
43. Software Maintenance & Software Project Measurement MCQ
44. Computer Architecture, Design, and Memory Technologies MCQ
45. Basic Structure of Computer MCQ
46. Computer Arithmetic MCQ
47. I/O Organization MCQ
48. Memory Organization MCQ
49. Multiprocessors MCQ
50. Introduction to Operating Systems MCQ
51. File Systems MCQ
52. CPU Scheduling MCQ
53. Memory Management MCQ
54. Input / Output MCQ
55. Operating Systems and Concurrency
56. Software Development and Architecture MCQ
57. Software architecture models MCQ

58. Software architecture implementation technologies MCQ
59. Software Architecture analysis and design MCQ
60. Software Architecture documentation MCQ
61. Introduction to Computational Intelligence MCQ
62. Fuzzy Systems MCQ
63. Genetic Algorithms MCQ
64. Rough Set Theory MCQ
65. Introduction to Swarm Intelligence, Swarm Intelligence Techniques MCQ
66. Neural Network History and Architectures MCQ
67. Autoencoder MCQ
68. Deep Learning MCQs
69. RL & Bandit Algorithms MCQs
70. RL Techniques MCQs
71. Review of traditional networks MCQ
72. Study of traditional routing and transport MCQ
73. Wireless LAN MCQ
74. Mobile transport layer MCQ
75. Big Data MCQ
76. Hadoop and Related Concepts MCQ
77. Hive, Pig, and ETL Processing MCQ
78. NoSQL MCQs Concepts, Variations, and MongoDB
79. Mining social Network Graphs MCQ
80. Mathematical Background for Cryptography MCQ
81. Cryptography MCQ
82. Cryptographic MCQs
83. Information Security MCQ
84. Cryptography and Information Security Tools MCQ

85. Data Warehousing MCQ
86. OLAP Systems MCQ
87. Introduction to Data& Data Mining MCQ
88. Supervised Learning MCQ
89. Clustering & Association Rule mining MCQ
90. Fundamentals of Agile Process MCQ
91. Agile Projects MCQs
92. Introduction to Scrum MCQs
93. Introduction to Extreme Programming (XP) MCQs
94. Agile Software Design and Development MCQs
95. Machine Learning Fundamentals MCQs
96. Neural Network MCQs
97. CNNs MCQ
98. Reinforcement Learning and Sequential Models MCQs
99. Machine Learning in ImageNet Competition mcq
100. Computer Network MCQ
101. Data Link Layer MCQ
102. MAC Sub layer MCQ
103. Network Layer MCQ
104. Transport Layer MCQ
105. Raster Scan Displays MCQs
106. 3-D Transformations MCQs
107. Visualization MCQ
108. Multimedia MCQs
109. INTRODUCTION Knowledge Management MCQs
110. Organization and Knowledge Management MCQs
111. Telecommunications and Networks in Knowledge Management MCQs

112. Components of a Knowledge Strategy MCQs
113. Advanced topics and case studies in knowledge management MCQs
114. Conventional Software Management MCQs
115. Software Management Process MCQs
116. Software Management Disciplines MCQs
117. Rural Management MCQs
118. Human Resource Management for rural India MCQs
119. Management of Rural Financing MCQs
120. Research Methodology MCQs
121. Research Methodology MCQs
122. IoT MCQs
123. Sensors and Actuators MCQs
124. IoT MCQs: Basics, Components, Protocols, and Applications
125. MCQs on IoT Protocols
126. IoT MCQs
127. INTRODUCTION Block Chain Technologies MCQs
128. Understanding Block chain with Crypto currency MCQs
129. Understanding Block chain for Enterprises MCQs
130. Enterprise application of Block chain MCQs
131. Block chain application development MCQs
132. MCQs on Service Oriented Architecture, Web Services, and Cloud Computing
133. Utility Computing, Elastic Computing, Ajax MCQs
134. Data in the cloud MCQs
135. Cloud Security MCQs
136. Issues in cloud computinG MCQs
137. Introduction to modern processors MCQs
138. Data access optimizations MCQs

139. Parallel Computing MCQs
140. Efficient Open MP Programming MCQs
141. Distributed Memory parallel programming with MPI MCQs
142. Review of Object Oriented Concepts and Principles MCQs.
143. Introduction to RUP MCQs.
144. UML and OO Analysis MCQs
145. Object Oriented Design MCQs
146. Object Oriented Testing MCQs
147. CVIP Basics MCQs
148. Image Representation and Description MCQs
149. Region Analysis MCQs
150. Facet Model Recognition MCQs
151. Knowledge Based Vision MCQs
152. Game Design and Semiotics MCQs
153. Systems and Interactivity Understanding Choices and Dynamics MCQs
154. Game Rules Overview Concepts and Case Studies MCQs
155. IoT Essentials MCQs
156. Sensor and Actuator MCQs
157. IoT Networking & Technologies MCQs
158. MQTT, CoAP, XMPP, AMQP MCQs
159. IoT MCQs: Platforms, Security, and Case Studies
160. MCQs on Innovation and Entrepreneurship
161. Innovation Management MCQs
162. Stage Gate Method & Open Innovation MCQs
163. Innovation in Business: MCQs
164. Automata Theory MCQs
165. Finite Automata MCQs

166. Grammars MCQs
167. Push down Automata MCQs
168. Turing Machine MCQs
169. Database Management System (DBMS) MCQs
170. Relational Data models MCQs
171. Data Base Design MCQs
172. Transaction Processing Concepts MCQs
173. Control Techniques MCQs
174. DBMS Concepts & SQL Essentials MCQs
175. DESCRIPTIVE STATISTICS MCQs
176. INTRODUCTION TO BIG DATA MCQ
177. BIG DATA TECHNOLOGIES MCQs
178. PROCESSING BIG DATA MCQs
179. HADOOP MAPREDUCE MCQs
180. BIG DATA TOOLS AND TECHNIQUES MCQs
181. Pattern Recognition MCQs
182. Classification Algorithms MCQs
183. Pattern Recognition and Clustering MCQs
184. Feature Extraction & Selection Concepts and Algorithms MCQs
185. Pattern Recognition MCQs
186. Understanding Cybercrime Types and Challenges MCQs
187. Cybercrime MCQs
188. Cyber Crime and Criminal justice MCQs
189. Electronic Evidence MCQs
190. Internet of Things MCQS
191. Analysis Design of Algorithm MCQ
192. Symmetric Key Cryptography MCQ

193. XML MCQs
194. System Security MCQs.
195. Linear Time- Invariant Systems mcqs
196. Control System MCQs: Basics, Feedback, and Analysis
197. OP-AMP applications MCQs
198. Radiation mcqs
199. NETWORKS mcqs
200. Satellite Services MCQs