Committee of Examiners for the Computer Science Test

Selected with the advice of the
Association for Computing Machinery (ACM) and the
Computer Society of the Institute of Electrical and Electronic Engineers (IEEE)

PROFESSOR JEAN-LOUP BAER, Chair
University of Washington

PROFESSOR STEPHEN COOK
University of Toronto

PROFESSOR RON K. CYTRON
Washington University

PROFESSOR DIRK C. GRUNWALD
University of Colorado

PROFESSOR VIJAYA RAMACHANDRAN
University of Texas at Austin

PROFESSOR JOHN A. TRIMBLE
Howard University

With the assistance of
J. R. Jefferson Wadkins and Mary Morley
Educational Testing Service

Practice Tests Available

GRE Subject Test practice books are available for each of the Subject Tests. Each book includes at least one test that was actually administered, answer sheets, correct answers, and data on how students who took the test performed on each question. Score conversion information is also provided to enable you to calculate your scaled score. Practice books may be ordered with a credit card (VISA, MasterCard, or American Express only) by calling 1-800-537-3160. Outside the U.S. and Canada, call 1-609-771-7243. Practice books may also be ordered on the registration form in the GRE Information and Registration Bulletin or from the GRE Web site at www.gre.org.

You may want to keep this booklet until after you receive your score report. It contains important information about content specifications on which your scores are based.

Copyright © 1999 by Educational Testing Service. All rights reserved.

EDUCATIONAL TESTING SERVICE, ETS, the ETS logo, GRADUATE RECORD EXAMINATIONS, and GRE are registered trademarks of Educational Testing Service. The modernized ETS logo is a trademark of Educational Testing Service.
Purpose of the GRE Subject Tests

The GRE Subject Tests are designed to help graduate school admission committees and fellowship sponsors assess the qualifications of applicants in specific fields of study. The tests also provide you with an assessment of your own qualifications.

Scores on the tests are intended to indicate knowledge of the subject matter emphasized in many undergraduate programs as preparation for graduate study. Because past achievement is usually a good indicator of future performance, the scores are helpful in predicting success in graduate study. Because the tests are standardized, the test scores permit comparison of students from different institutions with different undergraduate programs. For some Subject Tests, subscores are provided in addition to the total score; these subscores indicate the strengths and weaknesses of your preparation, and they may help you plan future studies.

The GRE Board recommends that scores on the Subject Tests be considered in conjunction with other relevant information about applicants. Because numerous factors influence success in graduate school, reliance on a single measure to predict success is not advisable. Other indicators of competence typically include undergraduate transcripts showing courses taken and grades earned, letters of recommendation, the GRE Writing Assessment score, and GRE General Test scores. For information about the appropriate use of GRE scores, write to GRE Program, Educational Testing Service, Mail Stop 51-L, Princeton, NJ 08541.

Preparing for a Subject Test

GRE Subject Test questions are designed to measure skills and knowledge gained over a long period of time. Although you might increase your scores to some extent through preparation a few weeks or months before you take the test, last-minute cramming is
unlikely to be of further help. The following information will help guide you if you decide to spend time preparing for the test.

- A general review of your college courses is probably the best preparation for the test. However, the test covers a broad range of subject matter, and no one is expected to be familiar with the content of every question.
- Use official GRE publications, published by ETS, to become familiar with questions used on the GRE Subject Tests. This descriptive booklet provides several sample questions. In addition, Subject Test practice books are available (see inside front cover).
- Become familiar with the types of questions used in the test, paying special attention to the directions. If you thoroughly understand the directions before you take the test, you will have more time during the test to focus on the questions themselves.

**Test-Taking Strategies**

The types of multiple-choice questions in the test are illustrated by the sample questions at the back of this booklet. When you take the test, you will mark your answers on a separate machine-scorable answer sheet. Total testing time is two hours and fifty minutes; there are no separately timed sections. Following are some general test-taking strategies you may want to consider.

- Read the test directions carefully, and work as rapidly as you can without being careless. For each question, choose the best answer from the available options.
- All questions are of equal value; do not waste time pondering individual questions you find extremely difficult or unfamiliar.
- You may want to work through the test quite rapidly, first answering only the questions about which you feel confident, then going back and answering questions that require more thought, and concluding with the most difficult questions if there is time.
- If you decide to change an answer, make sure you completely erase it and fill in the oval corresponding to your desired answer.
- Questions for which you mark no answer or more than one answer are not counted in scoring.
- As a correction for haphazard guessing, one-fourth of the number of questions you answer incorrectly is subtracted from the number of questions you answer correctly. It is improbable that mere guessing will improve your score significantly; it may even lower your score. If, however, you are not certain of the correct answer but have some knowledge of the question and are able to eliminate one or more of the answer choices, your chance of getting the right answer is improved, and it may be to your advantage to answer such a question.
• Record all answers on your answer sheet. Answers recorded in your test book will not be counted.
• Do not wait until the last five minutes of a testing session to record answers on your answer sheet.

**Development of the Subject Tests**

Each new edition of a Subject Test is developed by a committee of examiners composed of professors in the subject who are on undergraduate and graduate faculties in different types of institutions and in different regions of the United States and Canada. In selecting members for each committee, the GRE Program seeks the advice of the appropriate professional associations in the subject.

The content and scope of each test are specified and reviewed periodically by the committee of examiners. Test questions are written by the committee and by other faculty who are also subject-matter specialists and by subject-matter specialists at ETS. All questions proposed for the test are reviewed by the committee and revised as necessary. The accepted questions are assembled into a test in accordance with the content specifications developed by the committee to ensure adequate coverage of the various aspects of the field and at the same time to prevent overemphasis on any single topic. The entire test is then reviewed and approved by the committee.

Subject-matter and measurement specialists on the ETS staff assist the committee, providing information and advice about methods of test construction and helping to prepare the questions and assemble the test. In addition, each test question is reviewed to eliminate language, symbols, or content considered potentially offensive, inappropriate for major subgroups of the test-taking population, or likely to perpetuate any negative attitude that may be conveyed to these subgroups. The test as a whole is also reviewed to ensure that the test questions, where applicable, include an appropriate balance of people in different groups and different roles.

Because of the diversity of undergraduate curricula, it is not possible for a single test to cover all the material you may have studied. The examiners, therefore, select questions that test the basic knowledge and skills most important for successful graduate study in the particular field. The committee keeps the test up-to-date by regularly developing new editions and revising existing editions. In this way, the test content changes steadily but gradually, much like most curricula. In addition, curriculum surveys are conducted periodically to ensure that the content of a test reflects what is currently being taught in the undergraduate curriculum.

After a new edition of a Subject Test is first administered, examinees’ responses to each test question are analyzed in a variety of ways to determine whether each question functioned as expected. These analyses may reveal that a question is ambiguous,
requires knowledge beyond the scope of the test, or is inappropriate for the total group or a particular subgroup of examinees taking the test. Answers to such questions are not used in computing scores.

Following this analysis, the new test edition is equated to an existing test edition. In the equating process, statistical methods are used to assess the difficulty of the new test. Then scores are adjusted so that examinees who took a difficult edition of the test are not penalized, and examinees who took an easier edition of the test do not have an advantage. Variations in the number of questions in the different editions of the test are also taken into account in this process.

Scores on the Subject Tests are reported as three-digit scaled scores with the third digit always zero. The maximum possible range for all Subject Test total scores is from 200 to 990. The actual range of scores for a particular Subject Test, however, may be smaller. The maximum possible range of Subject Test subscores is 20 to 99; however, the actual range of subscores for any test or test edition may be smaller than 20 to 99. Subject Test score interpretive information is provided in Interpreting Your GRE Scores, which you will receive with your GRE Report of Scores.

What Your Scores Mean

Your raw score, that is, the number of questions you answered correctly minus one-fourth of the number you answered incorrectly, is converted to the scaled score that is reported. This conversion ensures that a scaled score reported for any edition of a Subject Test is comparable to the same scaled score earned on any other edition of the same Subject Test. Thus, equal scaled scores on a particular Subject Test indicate essentially equal levels of performance regardless of the test edition taken. Test scores should be compared only with other scores on the same Subject Test. (For example, a 680 on the Computer Science Test is not equivalent to a 680 on the Mathematics Test.)

Before taking the test, you may find it useful to know approximately what raw scores would be required to obtain a certain scaled score. Several factors influence the conversion of your raw score to your scaled score, such as the difficulty of the test edition and the number of test questions included in the computation of your raw score. Based on recent editions of the Computer Science Test, the table on the next page gives the range of raw scores associated with selected scaled scores for three different test editions. (Note that when the number of scored questions for a given test is greater than the range of possible scaled scores, it is likely that two or more raw scores will convert to the same scaled score.) The three test editions in the table that follows were selected to reflect varying degrees of difficulty. Examinees should note that future test editions may be somewhat more or less difficult than those test editions illustrated in the table.
Range of Raw Scores* Needed to Earn Selected Scaled Scores on Three Computer Science Test Editions That Differ in Difficulty

<table>
<thead>
<tr>
<th>Scaled Score</th>
<th>Raw Scores</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Form A</td>
<td>Form B</td>
<td>Form C</td>
</tr>
<tr>
<td>800</td>
<td>54-55</td>
<td>47-48</td>
<td>44-45</td>
</tr>
<tr>
<td>700</td>
<td>38-39</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>600</td>
<td>23</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>500</td>
<td>7-8</td>
<td>5-6</td>
<td>3-4</td>
</tr>
<tr>
<td>Number of Questions Used to Compute Raw Score</td>
<td>69</td>
<td>70</td>
<td>70</td>
</tr>
</tbody>
</table>

*Raw Score = Number of correct answers minus one-fourth the number of incorrect answers, rounded to the nearest integer.

For a particular test edition, there are many ways to earn the same raw score. For example, on the edition listed above as “Form A,” a raw score of 38 through 39 would earn a scaled score of 700. Below are a few of the possible ways in which a scaled score of 700 could be earned on that edition.

Examples of Ways to Earn a Scaled Score of 700 on the Edition Labeled As “Form A”

<table>
<thead>
<tr>
<th>Raw Score</th>
<th>Questions Answered Correctly</th>
<th>Questions Answered Incorrectly</th>
<th>Questions Not Answered</th>
<th>Number of Questions Used to Compute Raw Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>38</td>
<td>0</td>
<td>31</td>
<td>69</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>12</td>
<td>16</td>
<td>69</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>23</td>
<td>2</td>
<td>69</td>
</tr>
<tr>
<td>39</td>
<td>39</td>
<td>0</td>
<td>30</td>
<td>69</td>
</tr>
<tr>
<td>39</td>
<td>42</td>
<td>11</td>
<td>16</td>
<td>69</td>
</tr>
<tr>
<td>39</td>
<td>44</td>
<td>22</td>
<td>3</td>
<td>69</td>
</tr>
</tbody>
</table>
CONTENT OF THE COMPUTER SCIENCE TEST

The test consists of about 70 multiple-choice questions, some of which are grouped in sets and based on such materials as diagrams, graphs, and program fragments.

The approximate distribution of questions in each edition of the test according to content categories is indicated by the following outline. The percentages given are approximate; actual percentages will vary slightly from one edition of the test to another.

I. SOFTWARE SYSTEMS AND METHODOLOGY—35%
   A. Data organization
      1. Data types
      2. Data structures and implementation techniques
      3. File organization (e.g., sequential, indexed, multilevel)
   B. Program control
      1. Iteration and recursion
      2. Functions, procedures, and exception handlers
      3. Communication and synchronization
   C. Programming languages and notation
      1. Constructs for data organization and program control
      2. Scope, binding, and parameter passing
      3. Expression evaluation
   D. Systems
      1. Compilers and interpreters
      2. Operating systems, including resource management and protection/security
      3. Networking and distributed systems
      4. System development tools
      5. System performance
II. COMPUTER ORGANIZATION AND ARCHITECTURE—20%

A. Logic design
   1. Implementation of combinational and sequential circuits
   2. Functional properties of digital integrated circuits

B. Processors and control units
   1. Instruction sets
   2. Register and ALU organization
   3. Number representation
   4. Control sequencing
   5. Data paths

C. Memories and their hierarchies
   1. Speed, capacity, cost, allocation
   2. Cache, main, secondary storage
   3. Virtual memory, paging, segmentation

D. Communication
   1. Bus, switch, and network structures and protocols
   2. I/O
   3. Synchronization

E. High-performance architectures
   1. Pipelining super-scalar and out-of-order execution processors
   2. Parallel computing
   3. Distributed computing

III. THEORY—25%

A. Automata and language theory
   1. Models of computation (finite automata, pushdown automata, Turing machines)
   2. Formal languages (regular languages, context-free languages)
   3. Decidability

B. Design and analysis of algorithms and computational complexity
   1. Exact or asymptotic analysis of the best, worst, or average case for the time and space complexity of specific algorithms
   2. Algorithmic design techniques (divide and conquer, dynamic programming, greedy)
   3. Upper and lower bounds on the complexity of specific problems
   4. NP-completeness

C. Correctness of programs
   1. Formal specifications and assertions
   2. Verification techniques
IV. MATHEMATICAL BACKGROUND—15%

A. Discrete structures
   1. Mathematical logic
   2. Elementary combinatorics, including graph theory and counting arguments
   3. Elementary discrete mathematics, including number theory, discrete probability, recurrence relations

B. Numerical mathematics
   1. Computer arithmetic, including number representations, roundoff errors, overflow and underflow
   2. Classical numerical algorithms
   3. Linear algebra

V. ADVANCED TOPICS—5%
   Topics including modeling and simulation, information retrieval, artificial intelligence, computer graphics, data communications, databases, VLSI, logic programming.

SAMPLE QUESTIONS

The following questions are similar to those in the test. They illustrate the range of the actual test in terms of the abilities measured, the subject-matter areas tested, and the difficulty of the questions posed. An answer key appears after the sample questions.

Notations, Conventions, and Definitions:

In this test a reading knowledge of modern programming languages is assumed. The following notational conventions and definitions are used.

1. All numbers are assumed to be written in decimal notation unless otherwise indicated.

2. \( \lfloor x \rfloor \) denotes the greatest integer that is less than or equal to \( x \).

3. \( \lceil x \rceil \) denotes the least integer that is greater than or equal to \( x \).

4. \( g(n) = O(f(n)) \) denotes “\( g(n) \) has order at most \( f(n) \)” and means that there exist positive constants \( C \) and \( N \) such that \( 0 \leq g(n) \leq Cf(n) \) for all \( n > N \).

\( g(n) = \Omega(f(n)) \) denotes “\( g(n) \) has order at least \( f(n) \)” and means that there exist positive constants \( C \) and \( N \) such that \( g(n) \geq Cf(n) \geq 0 \) for all \( n > N \).
\( g(n) = \Theta(f(n)) \) denotes “\( g(n) \) has the same order as \( f(n) \)” and means that there exist positive constants \( C_1, C_2, \) and \( N \) such that
\[
0 \leq C_1 f(n) \leq g(n) \leq C_2 f(n) \quad \text{for all } n > N.
\]

5. In a logical expression:
\( \exists \) denotes “there exists.”
\( \forall \) denotes “for all.”
\( \rightarrow \) denotes “implies;” \( \supset \) and \( \Rightarrow \) are also used to denote “implies.”
\( \neg \) denotes “not”; “\( \overline{A} \)” is also used to mean “\( \neg A \)”.
\( \lor \) denotes “inclusive or”; + can also denote “inclusive or,” e.g., \( P + Q \) can denote “\( P \) or \( Q \).”
\( \oplus \) denotes “exclusive or.”
\( \land \) denotes “and”; also, juxtaposition of statements can denote “and,” e.g., \( PQ \) can denote “\( P \) and \( Q \).”
A boolean formula is satisfiable if it is true under some assignment of boolean values for its variables.
A boolean formula is a tautology (or is valid) if it is true under all assignments of boolean values for its variables.

6. \( \emptyset \) denotes the empty set.
If \( A \) and \( B \) denote sets, then:
\( A \cup B \) is the set of all elements that are in \( A \) or in \( B \) or in both;
\( A \cap B \) is the set of all elements that are in both \( A \) and \( B \);
\( \overline{A} \) is the set of all elements not in \( A \) that are in some specified universal set.

7. In a string expression, if \( S \) and \( T \) denote strings or sets of strings, then:
An empty string is denoted by \( \varepsilon \) or by \( \Lambda \);
\( ST \) denotes the concatenation of \( S \) and \( T \);
\( S + T \) denotes \( S \cup T \) or \( \{S, T\} \), depending on context;
\( S^n \) denotes \( S \cdot \ldots \cdot S \); \( n \) factors
\( S^* \) denotes \( \varepsilon + S + S^2 + S^3 + \ldots \);
\( S^+ \) denotes \( S + S^2 + S^3 + \ldots \).

8. In a grammar:
\( \alpha \rightarrow \beta \) represents a production in the grammar.
\( \alpha \Rightarrow \beta \) means \( \beta \) can be derived from \( \alpha \) by the application of exactly one production.
\( \alpha \Rightarrow^* \beta \) means \( \beta \) can be derived from \( \alpha \) by the application of zero or more productions.
Unless otherwise specified

(i) symbols appearing on the left-hand side of productions are nonterminal symbols, the remaining symbols are terminal symbols,
(ii) the leftmost symbol of the first production is the start symbol, and
(iii) the start symbol is permitted to appear on the right-hand side of productions.

9. In a logic diagram:

- represents an AND element.
- represents an inclusive OR element.
- represents an exclusive OR element.
- represents a NOT element.
- represents a NAND element.
- represents a NOR element.

10. input --------> \( D \) \( Q \) --------> clock represents a D-type flip-flop, which stores the value of its \( D \) input when clocked.

11. Binary tree traversal is defined recursively as follows:

- preorder - visit the root, traverse the left subtree, traverse the right subtree
- inorder - traverse the left subtree, visit the root, traverse the right subtree
- postorder - traverse the left subtree, traverse the right subtree, visit the root

12. In a finite automaton diagram, states are represented by circles, with final (or accepting) states indicated by two concentric circles. The start state is indicated by the word “Start.” An arc from state \( s \) to state \( t \) labeled \( a \) indicates a transition from \( s \) to \( t \) on input \( a \). A label \( ab \) indicates that this transition produces an output \( b \). A label \( a_1, a_2, \ldots, a_n \) indicates that the transition is made on any of the inputs \( a_1, a_2, \ldots, a_n \).

13. For a program segment \( S \) and predicates \( P \) and \( Q \), a triple \( \{ P \} S \{ Q \} \) is partially correct if, whenever \( P \) is true before initiation of \( S \), \( Q \) is true upon termination of \( S \). \( \{ P \} S \{ Q \} \) is totally correct if it is partially correct and \( S \) terminates for all inputs for which \( P \) is true.

Given that \( \{ P \} S \{ Q \} \) is partially correct, a precondition is any assertion that implies \( P \), and a postcondition is any assertion that is implied by \( Q \).
14. A loop invariant for a while statement

\[ \text{while } B \text{ do } S \]

is an assertion that is true each time the guard \( B \) is evaluated during execution of the while statement.

Directions: Each of the questions or incomplete statements below is followed by five suggested answers or completions. Select the one that is the best of the choices offered.

1. Which of the following arithmetic expressions corresponds directly to the parse tree given by the diagram in the figure above?

(A) \( 2(a - c) \)
(B) \( 2a - 2c \)
(C) \( 2(a + b - c + b) \)
(D) \( 2((a + b) - (c + b)) \)
(E) \( 2(a + b) - 2(c + b) \)

2. Which of the following statements must be true?

I. \( \lfloor x \rfloor = \lceil x \rceil \) if and only if \( x \) is an integer.
II. \( \lfloor x \rfloor + 1 = \lceil x \rceil \) if and only if \( x \) is not an integer.
III. \( \lfloor x + y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor \) for all \( x, y \).
IV. \( -\lfloor x \rfloor = \lceil -x \rceil \) for all \( x \).

(A) IV only    (B) I and IV only    (C) I, II, and III only
(D) I, II, and IV only    (E) I, II, III, and IV
3. The binary tree above can be used to store a sorted list so that an inorder tree walk generates the list in order. If the new entry \( K \) is to be made in such a way that alphabetical order is preserved, where should it be inserted?

(A) As left child of \( L \)
(B) As left child of \( N \)
(C) As right child of \( H \)
(D) As right child of \( R \)
(E) As right child of \( F \)

4. Many cryptographic protocols base their security on assumptions about the computational difficulty of integer factorization. Integer factorization serves this purpose because it is believed that

(A) integer multiplication is a function whose inverse, factorization, remains difficult for a large class of inputs
(B) \( P = NP \)
(C) even if \( P = NP \), integer factorization is still likely not to be polynomial-time computable
(D) testing primality is computationally intractable
(E) integer factorization is \( NP \)-hard
5. Consider a computer system in which processes can request and release one or more resources. Once a process has been granted a resource, the process has exclusive use of that resource until it is released. If a process requests a resource that is already in use, the process enters a queue for that resource, waiting until the resource is available. Which of the following will NOT deal effectively with the problem of deadlock?

(A) Giving priorities to processes and ordering the wait queues by priority
(B) Having a process request all its required resources when it first begins, and restarting if it cannot obtain them all
(C) Numbering the resources and requiring that processes request resources in order of increasing number
(D) Having processes time out and restart after a random interval of waiting
(E) Having the operating system monitor the wait queues and restart processes to break deadlocks

6. The graph above represents a finite state machine. Which of the following regular expressions describes the set of strings recognized by the finite state machine?

(A) 10*1+  (B) 10*1*  (C) 10*1  (D) (0 + 1)*  (E) 0*(0 + 1)*0*

7. In a language in which operations are associated right-to-left instead of left-to-right (i.e., $a + b + c = a + (b + c)$), the value of the expression

$$7 - (16/(3 + 1)*2) - 4$$

is

(A) $-1$  (B) 1  (C) 3  (D) 7  (E) 9
8. Assume that a list is constructed of elements with two fields, \textit{val} and \textit{next}, by using the \textit{next} field of each element to point to the next element in the list. If elements of the list are characterized in C by

\begin{verbatim}
typedef struct node *NodePtr;
struct node { int val;
                      NodePtr next; }
\end{verbatim}

which of the following is a correct implementation in C of the operation “insert \textit{p} after \textit{q}” where \textit{q} points to a list element and \textit{p} points to an element to be inserted?

(A) \textit{q}->next = \textit{p}->next ;
    \textit{p}->next = \textit{q} ;
(B) \textit{p}->next = \textit{q}->next ;
    \textit{q}->next = \textit{p} ;
(C) \textit{q}->next = \textit{p}->next ;
    \textit{p}->next = \textit{q}->next ;
(D) \textit{p}->next = \textit{q} ;
    \textit{q}->next = \textit{p}->next ;
(E) \textit{q}->next = \textit{p} ;
    \textit{p}->next = \textit{q}->next ;

9. Of the following, which best approximates the ratio of the number of nonterminal nodes to the total number of nodes in a complete \( K \)-ary tree of depth \( N \) ?

(A) \( \frac{1}{N} \)  \hspace{1cm} (B) \( N - \frac{1}{N} \)  \hspace{1cm} (C) \( \frac{1}{K} \) \hspace{1cm} (D) \( K = \frac{1}{K} \) \hspace{1cm} (E) \( \log_{10} \frac{1}{N} \)
10. Assume that the following table is the sequence of pages referenced by a program to be run in a 2-page memory. Assume demand paging is used.

<table>
<thead>
<tr>
<th>Time</th>
<th>Page Referenced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>stop</td>
</tr>
</tbody>
</table>

Which pages are in memory at time 6 if the program is run under the replacement rule LFU (the least frequently used page should be replaced)?

(A) 1 and 3  (B) 1 and 4  (C) 2 and 3  (D) 2 and 4  (E) 3 and 4

11. The table in the figure above shows the binary-coded-decimal (BCD) representation of the digits 0 through 9. The Boolean expression that represents the set of invalid codes is

(A) $A \lor BC$  (B) $ABCD$  (C) $AB \lor AD$  (D) $AB \lor CD$

(E) $AB \lor AC$
12. if $A > B$ then
   \[ V[I] := F(I) \]
else
   if $B > C$ then
   \[ V[I] := G(I) \]
Assume that the values of the Boolean expressions “$A > B$” and “$B > C$” are independent and that, on the average, $A > B$, 75 percent of the time and $B > C$, 25 percent of the time. If the program segment above is executed 10,000 times, how many times would one expect the functions $F$ and $G$ to be executed?

(A) $F: 2,500$, $G: 18,750$  (B) $F: 7,500$, $G: 625$  (C) $F: 7,500$, $G: 1,875$
(D) $F: 7,500$, $G: 2,500$  (E) $F: 9,375$, $G: 625$

13. Suppose one wishes to be certain that after execution of the statement
   \[ \text{if } a > b \text{ then } x := a; \]
the value of $x$ will equal the value of $a$.

Of the following, which is the weakest (least restrictive) condition that must necessarily hold before execution of that statement?

(A) $(x = a) \lor (a > b)$  (B) $(a > b) \land (x = a)$  (C) $a > b$
(D) $x > b$  (E) $x = a$

14. The function generated by the network in the figure above is

(A) $\overline{A} \overline{B} \overline{E} \lor EF \lor C \overline{D} \overline{F}$  (B) $(\overline{E} \lor A \overline{B} F)(C \lor D \lor \overline{F})$
(C) $(\overline{A} \overline{B} \lor E)(\overline{E} \lor \overline{F})(C \lor D \lor \overline{F})$  (D) $(A \lor B)\overline{E} \lor \overline{E} \overline{F} \lor C D \overline{F}$
(E) $(\overline{A} \lor \overline{B})E \lor EF \lor \overline{C} \overline{D} \overline{F}$
15. In the following function, $X$ is passed by reference and $Y$ is passed by value.

```pascal
function P(var X: integer; Y: integer): integer;
begin
  K := 3;
  L := 5;
  P := X + Y
end;
```

If the function $P$ were invoked by the following program fragment

```
K := 1;
L := 1;
Z := P(K, L)
```

then the value of $Z$ would be set equal to

(A) 2  (B) 3  (C) 4  (D) 6  (E) 8

16. In a certain implementation of strings, characters are 8 bits long and words are 36 bits long. Characters are packed into double words so that no space is lost. How many words of memory does a $k$-character string require? (If the string does not occupy an integral number of double words, enough binary zeros are appended to reach the next single word boundary.)

(A) $2^{k/9}$  (B) $k/9$  (C) $2k/9$  (D) $2k/9$  (E) $k/9$

17. A relation can be defined by giving the ordered pairs of elements for which the relation holds. If the relation $R$ over $\{a, b, c\}$ is given by $R = \{(a, a), (a, b), (b, a), (b, b), (c, c)\}$, which of the following properties does $R$ have?

I. Symmetry  II. Antisymmetry  III. Reflexivity  IV. Transitivity

(A) None  (B) II and III only  (C) II and IV only  (D) I, III, and IV  (E) II, III, and IV
18. The balanced ternary number system is a base-3 system in which the three digits are 0, 1, and \(-1\) (which is written as \(\bar{1}\)). The balanced ternary equivalent of the decimal number \(35 \frac{2}{9}\) is

(A) \(1\bar{1}11.01\)  (B) \(1\bar{1}1\bar{1}.1\)  (C) \(110\bar{1}.\bar{1}\)  (D) \(110\bar{1}.01\)  (E) \(110\bar{1}.\bar{1}\)

19. Assume that a data file has an index consisting of \(N\) items, where \(N\) is large. If a binary search of the index is used to find an item, then, of the following, which best approximates the mean number of comparisons required to locate a specific index entry?

(A) \(\frac{N + 1}{2}\)  (B) \(\frac{N(N - 1)}{2}\)  (C) \((\log_2 N) - 1\)

(D) \(N \log_2 N\)  (E) \(\frac{N + 1}{\log_2 N}\)

Questions 20-22 are based on the following program fragment.

\[
\begin{align*}
\text{(1)} & \quad \text{begin} \\
\text{(2)} & \quad \text{for } I := 1 \text{ to } N - 1 \text{ do} \\
\text{(3)} & \quad \text{for } J := I + 1 \text{ to } N \text{ do} \\
\text{(4)} & \quad \text{if } A[I] < A[J] \text{ then} \\
\text{(5)} & \quad \text{begin} \\
\text{(6)} & \quad \quad T := A[J]; \\
\text{(7)} & \quad \quad A[J] := A[I]; \\
\text{(8)} & \quad \quad A[I] := T \\
\text{(9)} & \quad \text{end} \\
\text{(10)} & \quad \text{end}
\end{align*}
\]

Assume that the program is syntactically correct, that all variables are of type \textit{integer}, that \(N\) has a value greater than 1, and that \(A\) is a one-dimensional array.
20. Upon entry to the program fragment above, the value of $N$ is 6 and the values of $A[1], \ldots, A[6]$ are given by:

<table>
<thead>
<tr>
<th>$A(1)$</th>
<th>$A(2)$</th>
<th>$A(3)$</th>
<th>$A(4)$</th>
<th>$A(5)$</th>
<th>$A(6)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-8$</td>
<td>$4$</td>
<td>$10$</td>
<td>$-2$</td>
<td>$7$</td>
<td>$3$</td>
</tr>
</tbody>
</table>

After exit from the program fragment, the values of $A[1], \ldots, A[6]$, respectively, will be

(A) 10, 7, 4, 3, $-2$, $-8$  (B) 10, $-8$, 7, 4, 3, $-2$  (C) 4, $-8$, $-2$, 10, 3, 7  
(D) $-2$, 3, 4, 7, $-8$, 10  (E) $-8$, $-2$, 3, 4, 7, 10

21. What is the greatest possible number of times in terms of $N$ that the block from line (5) through line (9) could be executed?

(A) $\frac{N(N - 1)}{2}$  (B) $N \log_2 N$  (C) $N^2$  (D) $\frac{N(N + 1)}{2}$  
(E) It cannot be determined from the information given.

22. If the order of statements (6) and (7) were reversed so that (6), (7), and (8) read:


(7) $T := A[J]$;

(8) $A[I] := T$

then the result of the program would be

(A) unchanged for all $A$ and $N$  (B) independent of the values of $A$  
(C) changed for all values of $N$  (D) unchanged for some values of $A$ and $N$  (E) unpredictable for some values of $A$ and $N$

23. A regular expression that denotes all strings of 0’s and 1’s that have at least two consecutive 1’s is

(A) $(0 + 10)^* 1(10 + 0)^*$  (B) $(0 + 10)^* 11(0 + 1)^*$  
(C) $(0 + 1)^* 10*1(0 + 1)^*$  (D) $0*11(0 + 10)^*(0 + 1)^*$  (E) $(0 + 11)^*$
24. Deduction in Prolog is based on the concept of unification. Two expressions $E$ and $F$ are said to be \textit{unifiable} if there are substitutions for the variables of $E$ and $F$ that make the expressions lexically identical. In the following three expressions, only $W$, $X$, $Y$, and $Z$ are variables.

\begin{enumerate}
  \item $f(W, W)$
  \item $f(X, 1)$
  \item $f(Y, g(Z))$
\end{enumerate}

Which of these expressions is (are) pairs of unifiable expressions?

(A) (I, II) only  
(B) (I, III) only  
(C) (II, III) only  
(D) (I, II) and (I, III) only  
(E) (I, II), (I, III), and (II, III)

25. If all variables are of type \textit{integer} and if $a \equiv b \pmod{m}$ and $x \equiv y \pmod{m}$, which of the following must be true?

\begin{enumerate}
  \item $a + x \equiv b + y \pmod{m}$
  \item $ax \equiv by \pmod{m}$
  \item $a/n \equiv b/n \pmod{m}$ for all $n \neq 0$
\end{enumerate}

(A) II only  
(B) III only  
(C) I and II only  
(D) I and III only  
(E) I, II, and III

26. Expressions in a certain language can be described in Backus-Naur form (BNF) as follows:

\begin{verbatim}
< expression > ::= < term > | < expression > < op1 > < term >
< term > ::= < item > | < term > < op2 > < item >
< item > ::= < variable > | < number >
\end{verbatim}

This syntax is most appropriate when the order of evaluation is

(A) from left to right always

(B) from left to right, but $op_1$ takes precedence over $op_2$

(C) from left to right, but $op_2$ takes precedence over $op_1$

(D) in any order, but $op_1$ takes precedence over $op_2$

(E) from right to left always
Questions 27-28 are based on the following operations on linear sequences of characters:

- **head**(x) = first character of x unless x = Λ, in which case **head**(x) is not defined.
- **tail**(x) = all of x except the first character. ( = Λ if x has < 2 characters.)
- **join**(x, y) = sequence that is the concatenation of x and y.

For example, if x = 'abc', then

- head(x) = 'a', tail(x) = 'bc', tail(head(x)) = Λ, and join(x, 'de') = 'abcde'

27. If x is a nonempty string, which of the following is necessarily true?
   (A) **join**(head(x), tail(x)) = x  (B) **head**(tail(x)) = tail(head(x))
   (C) **join**(head(x), x) = **join**(x, tail(x))  (D) **head**(tail(x)) = x
   (E) tail(head(x)) = x

28. Which of the following reverses the order of characters in a string of two or more characters if reverse(x) = x, where x is a string with a single character?
   (A) reverse(x) = **join**(reverse(tail(x)), head(x))
   (B) reverse(x) = **join**(tail(reverse(x)), head(x))
   (C) reverse(x) = **join**(tail(x), head(x))
   (D) reverse(x) = reverse(**join**(head(x), tail(x)))
   (E) reverse(x) = **join**(head(x), reverse(tail(x)))

```
S → AB
A → BB|0
B → AA|1
```

29. What is the number of terminal strings of length 5 generated by the context-free grammar shown above?
   (A) 4  (B) 5  (C) 6  (D) 7  (E) 8
Questions 30-32 are based on the diagram below, in which three 4-bit registers, a parallel adder, and four AND gates are used to implement the multiplication of two 4-bit, unsigned numbers. During the iterative part of the multiplication process, each clock pulse causes register $A$ to be parallel-loaded and, simultaneously, register $C$ to be right-shifted one bit position. The signal common to all four AND gates is the output from the rightmost bit position of register $C$. Clock and other control signals are not shown in the diagram.

30. For the system to be properly initialized, registers $A$, $B$, and $C$, respectively, should start out containing
   (A) multiplier, multiplicand, and zero   (B) multiplier, zero, and multiplicand
   (C) multiplicand, multiplier, and zero   (D) multiplicand, zero, and multiplier
   (E) zero, multiplicand, and multiplier

31. If the system has been properly initialized, the number of clock pulses required to complete the multiplication is
   (A) 3   (B) 4   (C) 7   (D) 8
   (E) dependent on the number of zeros in the multiplier
32. If appropriate circuitry for manipulating the sign bit were added, which of the following representations would allow the diagram above to be used in a circuit to multiply signed fixed-point numbers?

I. One’s complement
II. Two’s complement
III. Sign and magnitude

(A) None  (B) I only  (C) II only  (D) III only  (E) I, II, and III

33. If \( f \) denotes “the floating-point computation of,” which of the following must be true if it is assumed there is no overflow or underflow?

I. \( f(x + (y + z)) = f((x + y) + z) \)
II. If \( f(x + (y + z)) \) has an error, it is always a small relative error.
III. If \( f(x \cdot (y \cdot z)) \) has an error, it is always a small relative error.

(A) I only  (B) II only  (C) III only  (D) II and III only  (E) I, II, and III

34. Which of the following statements is(are) true?

I. There is a language \( L \) such that \( L \) is not recursive (\( L \) is undecidable), yet \( L \) and its complement are both recursively enumerable.
II. There is a language \( L \) such that \( L \) is not recursive, yet \( L \) is recursively enumerable.
III. Every language in \( NP \) is recursive.

(A) None  (B) II only  (C) I and II only  (D) II and III only  (E) I, II, and III
35. If \( T(0) = T(1) = 1 \), each of the following recurrences for \( n \geq 2 \) defines a function \( T \) on the nonnegative integers. Which of the following CANNOT be bounded by a polynomial function?

(A) \( T(n) = 3T\left[\lfloor n/2 \rfloor\right] + n^2 \)

(B) \( T(n) = 4T\left[\lfloor n/2 \rfloor\right] + n \)

(C) \( T(n) = T\left[\lfloor n/3 \rfloor\right] + 8n + 1 \)

(D) \( T(n) = 2T(n - 2) + 1 \)

(E) \( T(n) = T(n - 1) + n^2 \)

---

**Answer Key**


