Contents

Preface *xix*
Acknowledgements *xxi*

1 ARRAY—*EASY, CONTIGUOUS, ELEGANT!* **1**

Introduction *1*
1.1 How to Initialize an Array *1*
1.2 How to Traverse an 1D Array using Index *2*
1.3 How to Manipulate Elements of the Array *3*
1.4 How to Add Array Elements in a Specific Region *4*
1.5 How to Add Elements in the Odd and Even Places in the Array *4*
1.6 How to Perform Operations Involving External Variables *5*
1.7 How to Find Function Values *7*
1.8 How to Solve a Demographical Application, a Problem of Vital Statistics *7*
1.9 Where to Apply 3D Arrays *8*
1.10 How to Delete a Particular Item from an Array *9*
1.11 How to Delete an Item from a Particular Location *10*
1.12 How to Find the Maximum Number in an Array *11*
1.13 How to Find the Minimum Number in an Array *12*
1.14 How to Sort the Array Alphabetically *12*
1.15 How to Check If a String is a Palindrome or not *13*
1.16 How to Search for an Array Element *14*
1.17 How to Make the Array Elements Unique *14*
1.18 How to Find the Mean of the Array Elements *16*
1.19 How to Find Weighted Average of an Array of Numbers *16*
1.20 How to Find the Median of the Array Elements which are Already Sorted *17*
1.21 How to Find the Mode of the Array Elements *17*
1.22 How to Find the Range of the Array Elements *18*
1.23 How to Find Standard Deviation of an Array *18*
1.24 How to Find the Variance of the Array Elements *19*
1.25 How to Find an Interpolated Value using Newton's Forward Difference Interpolation *20*
1.26 How to Interpolate using Lagrange's Interpolation Formula *21*
1.27 How to Find a Regression Line on *X* or *Y* *23*
1.28 How to Find Simple Aggregation Index Number *24*
1.29 How to Find the Simple Average of a Price-Relative Index *25*
1.30 How to Find Laspeyre's Index Number *26*
1.31 How to Find Paasche's Index Number *26*
1.32 How to Find Bowley's Index Number *26*
1.33 How to Find Fisher's Index Number *27*
1.34 How to Find Marshall–Edward Index Number *27*
1.35 How to Represent a Matrix Using 2D Arrays *27*

1.36 How to Add Two 3 × 3 Matrices *27*
1.37 How to Subtract Two 3 × 3 Matrices *28*
1.38 How to Multiply Two Matrices *28*
1.39 How to Calculate Revenues using Matrix Multiplication *30*
1.40 Multiplication of Two 2 × 2 Matrices using Strassen's Algorithm which uses 7 Multiplications and 18 Additions *30*
1.41 How to Find the Hadamard Product of Two Matrices *31*
1.42 How to Find the Kronecker Product of Two Matrices *32*
1.43 How to Find the Transpose of a Matrix *34*
1.44 How to Find the Inverse of a Square Matrix *34*
1.45 How to Find the Upper Triangular of a Matrix *35*
1.46 How to Find a Strict Upper Triangular Matrix *36*
1.47 How to Find the Lower Triangular of a Matrix *37*
1.48 How to Find a Strict Lower Triangular Matrix *37*
1.49 How to Create a Toeplitz Matrix from a given Row and Column *38*
1.50 How to Find whether a Matrix is Symmetric or not *39*
1.51 Representing a Sparse Matrix as Arrays *39*
1.52 3D Array Applications How to use a 3D Array to Store and Manipulate the Literacy Details of 5 Cities around a Year *42*
1.53 How to Return More than one Value from a Function *43*
1.54 How to Clone String Tokenizer Class of Java *44*
1.55 Conversion of Binary to Decimal *44*
1.56 How to Design a Chart for Share Trading *45*
1.57 How to Find HHI Index *46*
1.58 How to Find GINI Coefficient Measurement for a City *47*
1.59 How to Find whether Three given Numbers are in AP, GP or HP *48*
1.60 Animation of Different Signaling Formats *49*
1.61 A Well-known Cryptographic Technique—Cipher Text *54*
1.62 Decoder Program for the above Encrypter *55*
1.63 How to Find the Histogram of a 256 Gray Scale Image *55*
1.64 How to Convert a Gray Scale Image to Binary Image/Negative Image *56*
Revision of Concepts *57*
Review Questions *58*
Programming Problems *59*

2 STRUCTURES—*THE BUILDING BLOCKS* **61**

Introduction *61*
2.1 Use of `typedef` *61*
2.2 Accessing the Structure Elements *62*
2.3 Some Built-in useful Structures in Turbo C (under DOS) *63*
2.4 How to Define a Structure that Represents a Point in 3D *63*
2.5 How to Find the Centroid of a Polygon using Point Structure *64*
2.6 How to Find the Distance between Two Points in 3D *64*
2.7 How to Find the Area of Any Regular Polygon *65*
2.8 How to Test Collinearity for Three Points *65*
2.9 How to Check If a Triangle is Equilateral *66*
2.10 How to Check If a Triangle is Isosceles *66*
2.11 How to Model a Triangle using Point Structure? *66*
2.12 How to Check If a Triangle is Right Angled *67*
2.13 How to Find whether a Triangle is Equilateral or not *67*
2.14 How to Model a Tetrahedron using Triangles *68*

2.15 How to Model a Rectangle using `Struct` and `Enum` *68*
2.16 How to Model a Trapezium using Point *69*
2.17 How to Check whether a Trapezium is Equilateral or not *69*
2.18 How to Find whether a Point is within a Triangle or not *70*
2.19 How to Find whether a Point is within a Rectangle or not *71*
2.20 How to Find whether a Point is within a Circle or not *71*
2.21 How to Find whether Two Circles are Touching Internally or not *71*
2.22 How to Find whether Two Circles are Touching Externally or not *72*
2.23 How to Model a Straight Line in Slope Format *72*
2.24 How to Model a Straight Line in *XY* Intercept Format *72*
2.25 How to Convert an *XY* Intercept form Line to Slope Format Line *73*
2.26 How to Convert a Slope Line Format to *XY* Intercept Format *73*
2.27 How to Find whether Two Lines are Parallel or not *73*
2.28 How to Find the Point of Intersection of Two Straight Lines *73*
2.29 How to Find the Tangent on any Point on a Circle *73*
2.30 How to Model a Parabola using a Straight Line and Point *74*
2.31 How to Find the Tangent on any Point on a Parabola *74*
2.32 How to Find the Normal on any Point on a Parabola *74*
2.33 How to Model an Ellipse *74*
2.34 How to Find the Area of an Ellipse *75*
2.35 How to Find the Tangent at any Point of an Ellipse *75*
2.36 How to Find the Normal at any Point of an Ellipse *75*
2.37 How to Model a Prism using Structure *75*
2.38 How to Model a Circular Cylinder *76*
2.39 How to Find the Surface Area of a Cylinder *76*
2.40 How to Model a Cone *77*
2.41 How to Find the Area of a Cone *77*
2.42 How to Find the Volume of the Cylinder Defined by a Circle and Point *77*
2.43 How to Find the Area of the Prism *78*
2.44 How to Find out whether a Point is within an Ellipse or not *78*
2.45 How to Find out whether a Point is within a Hyperbola or not, Assume that the Major or Minor Axes are Given *78*
2.46 How to Model a Rhombus *79*
2.47 How to Find the Area of a Rhombus *79*
2.48 How to Model Vectors as Structure *79*
2.49 How to Write a Function to Add Vectors *79*
2.50 How to Find the Weighted Sum of Vectors *80*
2.51 How to Find If the Weighted Sum of Vectors is an Affine Summation or not *80*
2.52 How to Write a Function to Find DOT Product of Two Vectors *81*
2.53 How to Write a Function to Find Cross Product of Two Vectors *81*
2.54 How to Write a Function for Scalar Multiplication of a Vector *82*
2.55 How to Find Dot Product of Three Vectors *82*
2.56 How to Find whether Three Vectors are Coplanar or not *82*
2.57 How to Find the Cross Product of Three Vectors *82*
2.58 How to Find the Scalar Product of Four Vectors *83*
2.59 How to Find the Vector Product of Four Vectors *83*
2.60 How to Model a Complex Number as a Structure *83*
2.61 How to do Conversion from Polar to Rectangular Form and vice versa *84*
2.62 How to Add Complex Numbers *84*
2.63 How to Subtract One Complex Number from another *85*

2.64 How to Write a Function to Multiply Two Complex Numbers *85*
2.65 Proving De Moivre's Theorem using Polar Complex Structure *86*
2.66 How to Write a Phonebook Simulation Program using Structure *86*
2.67 How to Model a Bank Account as a Combination of Structures *89*
2.68 How to Write a POS (Point of Sale) Simulation using Structure *91*
Revision of Concepts *102*
Review Questions *103*
Programming Problems *103*

3 LINKED LIST—*SCATTERED YET LINKED!* **105**

Introduction *105*
3.1 Single Linked List *106*
3.2 Double Linked List *107*
3.3 Circular Linked List *108*
3.4 What do you Mean by Array of Linked Lists? *108*
3.5 Linked List in C and Predictors *109*
3.6 Linked List Function Philosophy of this Chapter *109*
3.7 How to Insert a Node at the end of a Single Linked List—The Node may be the First Node of the Linked List *111*
3.8 How to Insert a Node at the Front of the Single Linked List *111*
3.9 How to Find the Front Element of the Single Linked List *112*
3.10 How to Find the Back Element of the Single Linked List *112*
3.11 How to Traverse the Single Linked List *113*
3.12 How to Count the Number of Nodes in a Single Linked List *113*
3.13 How to Find the Frequency of an Item in a Single Linked List *113*
3.14 How to Search a Particular Item in the Single Linked List *114*
3.15 How to Find the Address of a Particular Node in a Single Linked List *114*
3.16 How to Insert Nodes at a Particular Location in a Single Linked List *115*
3.17 How to Insert Nodes before a Particular Node in a Single Linked List *116*
3.18 How to Display all the Contents of a Single Linked List *116*
3.19 How to Find the Maximum Element from a Single Linked List *116*
3.20 How to Find the Minimum Element from a Single Linked List *117*
3.21 How to Edit the Content of a Particular Node with a Given Value *117*
3.22 How to Write a Function to Merge Two Linked Lists *118*
3.23 How to Write a Function to Insert a List within Another List *119*
3.24 How to Swap the Head and Tail Node (i.e. The First and the Last Node) of the Single Linked List *120*
3.25 How to Swap the Contents of Any Two Other Nodes apart from Head and Tail *121*
3.26 How to Delete a Particular Node given by an Index Number *121*
3.27 How to Delete a Range of Elements from a List *122*
3.28 How to Delete Alternate Elements from a Single Linked List *122*
3.29 How to Make the List Entries Unique *123*
3.30 How to Delete the First Element of the List *123*
3.31 How to Delete the Last Element of the List *124*
3.32 Linked List and Predicates *124*
3.33 What are the Attributes and Methods of a Polynomial as a Data Structure? *125*
3.34 How to Represent a Polynomial using a Single Linked List *126*
3.35 Polynomial Tool Box *126*
3.36 How to Add a New Term to a Polynomial *127*
3.37 How to Add Two Polynomials and Return Their Sum *128*

3.38 How to Multiply Two Polynomials *128*
3.39 How to Find the Differentiation of a Polynomial *128*
3.40 How to Calculate the Integral of a Polynomial *128*
3.41 How to Evaluate the Value of the Polynomial at a Value *129*
3.42 How to Find the Definite Integral Value of a Function *129*
3.43 How to Display a Polynomial *129*
3.44 How to Find the Value of a Composite Function *129*
3.45 How to Add a New Term to a Polynomial *130*
3.46 How to Add Two Polynomials of Three Variables *131*
3.47 How to Multiply Two Polynomials of Three Variables *131*
3.48 How to Differentiate a Polynomial with Respect to *x* *132*
3.49 How to Differentiate a Polynomial with Respect to *y* *132*
3.50 How to Differentiate a Polynomial with Respect to *z* *132*
3.51 How to Integrate the Polynomial with Respect to *x* Assuming the Other Two Variables are Keeping Constant *132*
3.52 How to Integrate a Polynomial with Respect to *y* Assuming the Other Two Variables are Keeping Constant *133*
3.53 How to Integrate a Polynomial with Respect to *z* Assuming the Other Two Variables are Keeping Constant *133*
3.54 How to Integrate a Polynomial when All Three Variables are Varying *133*
3.55 How to Evaluate a Polynomial at Given Values of *x*, *y* and *z* *133*
3.56 How to Integrate a Polynomial within a Given Limit *134*
3.57 Some Applications of Polynomial Toolbox *134*
3.58 How to Find the Curl of a Function of Three Variables *134*
3.59 Huge Numbers: Application of Linked Lists *135*
3.60 How to Store Two Huge Integers as Single Linked List and then Add those Two Numbers and Display the Summation *136*
3.61 Digital Signal Processing *138*
3.62 How to Find the Length of a Signal *138*
3.63 How to Find the Index of a Given Amplitude in a Signal *139*
3.64 How to Add a New Value at the End of a Signal *139*
3.65 How to Add a New Value at the Front of a Signal *140*
3.66 How to Return the First Signal Node Pointer *140*
3.67 How to Return the Last Signal Node Pointer *141*
3.68 How to Insert a Node at a Particular Location of a Signal *141*
3.69 How to Display a Digital Signal *141*
3.70 How to Get the Amplitude of the First Signal Node *142*
3.71 How to Get the Amplitude of the Last Signal Node *142*
3.72 How to Find Frequency of a Particular Amplitude in a Given Signal *142*
3.73 How to Get the Address of a Signal Node Given the Index *142*
3.74 How to Get the Amplitude of a Signal at a Particular Point *143*
3.75 How to Check whether a Digital Signal is Even or Not *143*
3.76 How to Check whether a Digital Signal is Causal *143*
3.77 How to Check whether a Digital Signal is Anti-Causal *144*
3.78 How to Check whether a Digital Signal is Non-Causal *145*
3.79 How to Add at the End of a Single Circular Linked List *146*
3.80 Double Linked List *146*
3.81 How to Add a Number at the End of a Double Linked List *147*
3.82 How to Add a Number at the Front of a Double Linked List *147*
3.83 How to Go to the Next Node of a Double Linked List *147*

3.84 How to Go to the Previous Node of a Double Linked List *148*
3.85 How to Display the Double Linked List in Forward Direction *148*
3.86 How to Display the Double Linked List in Backward Direction *148*
3.87 How to Insert a Value at a Location in the Linked List *148*
3.88 How to Add a Number at the end of a Circular Doubly Linked List *149*
3.89 Linked List Applications in Biochemistry *149*
3.90 How to Hybridize Two Single DNA Strands to One DNA *151*
3.91 How to Melt One DNA to a Couple of Strands *152*
3.92 How to Emulate Linking of One DNA Strand to the Other *153*
3.93 How to Represent a Sparse Matrix using Jagged Arrays *153*
3.94 How to Add an Item in the Sparse Matrix *154*
3.95 How to Add a Jagged Row to the Jagged Representation of the Sparse Matrix *155*
3.96 How to Accept the Sparse Matrix Details from the User and Create the Jagged Array of Linked Lists to Represent the Matrix in a Space-efficient Way *156*
3.97 Representation of Handwritten Signatures using Jagged Arrays *157*
3.98 How to Model *Simple Content Management* System using Linked List *157*
3.99 How to Model *Workflow Engine* System (*Like* K2.NET) using Linked List *161*
3.100 A Comparison between Arrays and Linked Lists *161*
Revision of Concepts *163*
Review Questions *163*
Programming Problems *164*

4 STRINGS—*DATABASE TO DNA!* **165**

Introduction *165*
4.1 Some Key Facts about Strings in C *165*
4.2 C-Style String *166*
4.3 How to Initialize at the Time of Declaration *166*
4.4 How to Initialize Strings using user-Defined Values *166*
4.5 How to Initialize One String with Another String *167*
4.6 How to Initialize a String using Character Values *167*
4.7 How to Initialize a String using ASCII Values *167*
4.8 Introduction to some Built-in Turbo C String Library Functions *168*
4.9 Designing Utility Tools using these Two Functions *172*
4.10 A Tool for Changing Case of Few Chosen Abbreviations in a File (Using strupr()) *173*
4.11 How to Reverse a String *174*
4.12 How to Set Characters of a String with Another Character *175*
4.13 How to Find the First Occurrence of a Character of a Substring within Another String *176*
4.14 How to Find the Location from where Two Strings Start to Differ *177*
4.15 How to Create the Duplicate of a String in a Memory-Efficient Manner *178*
4.16 How to Tokenize a given String *178*
4.17 What do you mean by Prefix of a String? *179*
4.18 What do you mean by Suffix of a String? *180*
4.19 What do you mean by Subsequence of a String? *181*
4.20 How to Check whether a String is a valid ISBN or not *194*
4.21 How to Check Validity of a Social Insurance Number (SIN) Code *195*
4.22 How to Check whether a given Credit Card Number is Valid or not *195*
4.23 How to Change the Case of a Sentence to Sentence Case *200*
4.24 How to Toggle the Case of the Letters of a Sentence *200*
4.25 How to Find out the Soundex Code for a given Word *205*
Review Questions *207*
Programming Problems *207*

5 RECURSION—*TIME AND AGAIN!* **208**

Introduction *208*
5.1 Different types of Recursion *208*
5.2 Pitfalls of Recursion.. *209*
5.3 Fibonacci Numbers and Golden Ratio *209*
5.4 Random Number Generation using Recursion *212*
5.5 How to Generate Pseudo Random Numbers(PRNs) using Von Numann's Middle Squaring Method *213*
5.6 How to Generate the Ackermann's Function *216*
5.7 What is Inverse Ackermann's Function? *216*
5.8 How to Generate TAK Function for given Variables *216*
5.9 Solving Non-Linear Equations using Recursion *221*
5.10 Pattern Generation using Recursion *229*
5.11 How to Write a Recursive Function to Generate the Numbers of the Pascal Triangle *232*
5.12 What is the Relationship between Pascal Triangle Numbers and Fibonacci Numbers? *233*
5.13 How to Write a Recursive Function to Generate the Numbers of the Bell Triangle. To Accept Two Numbers, One for Row and the Other for Column *233*
5.14 Application of Bell Numbers *234*
5.15 How to Write a Recursive Function to Generate the Numbers of Bernoulli Triangle *234*
5.16 How to Write a Recursive Function to Generate the Numbers of Catalan's Triangle *235*
5.17 What is the Recursive Relation that Generates a Catalan Number? *236*
5.18 Solving Euler's Polygon Division using a Catalan Number *236*
5.19 DYCK Path and Catalan Number *236*
5.20 Ballot Problem and Catalan Number *237*
5.21 How to Write a Recursive Function to Generate the Numbers of Losanitsch's Triangle *237*
5.22 How to Write a Recursive Function to Generate the Numbers of the Leibnitz Harmonic Triangle *238*
5.23 L-System, Recursion and more Fractals *242*
5.24 Fractal Generation using Recursion *243*
5.25 Koch Curve *243*
5.26 Koch Snowflake *243*
5.27 Recursion in Natural Scene Generation *244*
Revision of Concepts *245*
Review Questions *245*
Programming Problems *246*

6 STACK—*ONE UPON ANOTHER* **247**

Introduction *247*
6.1 Model a Stack as a Struct *247*
6.2 How to Initialize the Stack Modeled above *248*
6.3 How to Pop The MRA Element from the above Stack *249*
6.4 How to Display the Stacktop Element *249*
6.5 How to Swap the Top Two Elements *250*
6.6 Putting it All Together using Arrays *250*
6.7 Model a Stack using a Linked List *251*
6.8 Pushing an Element *252*
6.9 How to Pop an Element from the Stack *252*
6.10 How to Peep at the Stack Top *253*
6.11 How to Swap the Top Two Elements *253*
6.12 How to Write a Parenthesis Matcher using Stack *258*
6.13 Switchbox Routing Problem *258*

6.14 Saguaro Stack *265*
6.15 How to Write the Algorithm to use a Saguaro Stack to Check a Wrongly Entered URL *267*
6.16 What is an MTFL? *271*
6.17 How to Model an MTFL using Two Stacks which are themselves Modeled by a Linked List *271*
6.18 How to Find the Most Sought Item in a Departmental Store using an MTF List *276*
6.19 What is Backtracking? *277*
6.20 How to Develop a Backtracking Algorithm to Find a Path in a Maze using Stack *277*
Revision of Concepts *282*
Review Questions *282*
Programming Problems *283*

7 QUEUE—*WAITING OR PRIVILEGED?* **284**

Introduction *284*
7.1 How to Model a Linear Queue using an Array *284*
7.2 How to Initialize the Linear Queue Defined above *285*
7.3 How to Append an Element in the Queue *285*
7.4 How to Delete from a Queue *286*
7.5 How to Search an Element in the Queue *288*
7.6 How to Display the Elements in a Queue *289*
7.7 Model a Queue using a Linked List *289*
7.8 How to Append an Item to a Linked Queue *290*
7.9 How to Delete the Front Item of a Queue *291*
7.10 How to Search an Element in a Queue *291*
7.11 How to Display the Elements of a Queue *292*
7.12 Model a Linear Queue using two Stacks *294*
7.13 How to Model a Stack using two Queues *295*
7.14 Model a Circular Queue using Structure *297*
7.15 Model a Priority Queue using an Array *298*
7.16 Model a Priority Queue using a Single Linked List *302*
7.17 An Application of Priority Queue—*Scheduling Appointments* *309*
7.18 How to Model a Deque (Double-Ended Queue) using a Linked List *316*
7.19 How to Model a Move to Front List (MTFL) using a Queue *317*
7.20 How to Simulate the Queue in Front of the Cash Counter *318*
Revision of Concepts *322*
Review Questions *323*
Programming Problems *324*

8 TREES—*EXPLORER TO GENETICS!* **325**

Introduction *325*
8.1 What are the Different Ways Trees are Represented? *327*
8.2 What is a Strictly Binary Tree? *327*
8.3 What is an almost Complete Binary Tree? *327*
8.4 What is a Complete Binary Tree? (*Also known* as Perfect Binary Tree) *327*
8.5 What is a Weak Binary Tree? *328*
8.6 What is a Strong Binary Tree? *328*
8.7 How to Model a Binary Tree using an Array *329*
8.8 How to Find whether a Node is a Right Child of its Parent *333*
8.9 How to Find the Address of the Sibling of a Node *335*
8.10 How to Find the Address of the Uncle of a Node *335*
8.11 How to Traverse the Tree "In-order" *335*

8.12 What is a Binary Search Tree? *337*
8.13 How to Search a Value in a BST *340*
8.14 What is the Right Rotation on a BST? *341*
8.15 Some Areas of Application of BST *342*
8.16 What is an Expression Tree? *343*
8.17 What is a Decision Tree? *347*
8.18 Binary Search Tree and Games *348*
8.19 How to Handle Multiple Subjects in such an Adaptive Test *351*
8.20 How to Convert a Multiway Tree to a Binary Tree *353*
8.21 What is the Balance Factor of a BST? *354*
8.22 How to Balance a Binary Search Tree *355*
8.23 What is a Splay Tree? *356*
8.24 What is a Heap? *358*
8.25 What are the Different Approaches to Create a Heap? *358*
8.26 How to Implement a Binary Heap using Array *359*
8.27 What is an AVL Tree? *359*
8.28 How to Insert an Element to an AVL Tree *359*
8.29 What is a BSP Tree? *360*
8.30 What is a Quad Tree? A 2D Variation of BSP (Also known as Q-tree)? *360*
8.31 How to Get the North-East Uncle of a Quad Tree Node *362*
8.32 How are Images Represented using a Quad Tree? *364*
8.33 How to Convert a Quad Tree to a Binary Tree *365*
8.34 Superimposing Multiple Binary Images using a Binary Tree *365*
8.35 Handwriting Recognition using Quad Tree *367*
8.36 How to Compress Images using a Quad Tree *368*
8.37 What is an Octree? *370*
8.38 What is a Trie? *371*
8.39 How to Model a Trie using Linked List *371*
8.40 How to Add a Key to a Trie *372*
8.41 How to Search a Key in a Trie *372*
8.42 How to Find whether a Key in a Trie can be Deleted or Not *375*
8.43 How to use a Trie for Spell Checking *376*
Review Questions *377*
Programming Problems *377*

9 GRAPHS—*MATHEMATICS TO WAN* ***378***

Introduction *378*
9.1 What are the Different Ways Graphs are Represented? *378*
9.2 How to Add an Edge in a Graph Modeled by Adjacency Matrix *379*
9.3 How to Remove an Edge in a Graph Modeled by Adjacency Matrix *379*
9.4 What is a Path Matrix? *379*
9.5 How to Find whether a Graph is a Tree or not *379*
9.6 What is Minimum Spanning Tree of a Graph? *380*
9.7 Prim's Algorithm *380*
9.8 Kruskal's Algorithm to find the MST *381*
9.9 Reverse Delete Algorithm to find the MST *382*
9.10 What is a Directed Acyclic Graph or DAG? *384*
9.11 What is Topological Sorting of a DAG? *384*
9.12 How to Find whether a Graph is Planer or not *392*

9.13 Breadth First Search (BFS) *396*
9.14 Depth First Search (DFS) *398*
Revision of Concepts *398*
Review Questions *399*
Programming Problems *399*

10 SORTING—*MICRO, MACRO, MAMMOTH* ***400***

Introduction *400*
10.1 Functions in this Chapter *400*
10.2 Sorting Algorithms Classifications *400*
10.3 Exchange Sort Algorithms *400*
10.4 What is the Time Complexity of Bubble Sort? *403*
10.5 What is Odd-Even Transposition Sort? *404*
10.6 What is the Time Complexity of Bidirectional Bubble Sort? *406*
10.7 What is the Time Complexity of Comb Sort? *409*
10.8 Insertion Sort Algorithms *410*
10.9 What is the Time Complexity of Insertion Sort? *410*
10.10 Comparison with the Ideal *O*(*N*^2) Curve *411*
10.11 What is the Time Complexity of Binary Insertion Sort? *413*
10.12 Problems with Insertion Sort: Shifting *413*
10.13 Explain the Library Sort Algorithm (*Also known as* Gapped Insertion Sort) *414*
10.14 What is the Time Complexity of Shell Sort? *415*
10.15 Selection Sort Algorithms *416*
10.16 What is the Time Complexity of Selection Sort? *417*
10.17 What is Bingo Sort? *418*
10.18 Hybrid Sort Algorithms *418*
10.19 What is J-Sort? *418*
10.20 Divide-N-Conquer Sorting Algorithms *419*
10.21 How to Write a Function to Demonstrate Quick Sort *419*
10.22 What is the Time Complexity of Quick Sort? *419*
10.23 How to Select the Pivot in Quick Sort *420*
10.24 What is the Time Complexity of Merge Sort? *421*
10.25 What is the Time Complexity of Stooge Sort? *422*
10.26 Distribution Sorting Algorithms *422*
10.27 Bucket Sort *422*
10.28 Performance Comparisons of the Sorting Algorithms with *O* (*n*^2) Time Complexity *423*
10.29 Performance Comparisons of the Sorting Algorithms with *O* (*n* log *n*) Time Complexity *424*
10.30 Bogo Sort and Friends *424*
10.31 Tree Sort *425*
10.32 Lexicographic Sort *425*
10.33 Radix Sort *426*
10.34 Address Calculation Sort using Hashing *426*
10.35 Application of Sorting *428*
10.36 What is Clustering? *428*
10.37 Business Clustering *428*
10.38 Finding the Shortest Path *428*
10.39 Finding the Most Wanted DVD in the City *428*
10.40 Finding the Greatest Online Scorer in the Online Pool Competition *428*
10.41 Finding the Largest Shape when their Dimensions are Given *429*
Revision of Concepts *429*
Review Questions *430*
Programming Problems *430*

11 HASHING—*ACCIDENT OR CHOICE?* **431**

Introduction *431*
11.1 Concept of Collision and its Resolution *431*
11.2 Some Key Facts and Jargons about Hashing *433*
11.3 How to Demonstrate the Separate Chaining Method for Hashing Elements in a Hash Table *435*
11.4 What is Coalesced Hashing? *437*
11.5 What are the Variations of Coalesced Hashing (*Linked Hashing*)? *441*
11.6 What is a Hash Chain and What is its Utilization for OTP? *441*
11.7 How is a Hash Tree used to Check the Data Integrity of a Media Downloaded from a Peer-to-Peer (P2P) Network *443*
Review Questions *444*
Programming Problems *444*

12 ADT—*DELIVERED INBUILT PLUMBING!* **445**

12.1 The Black-Box Concept *445*
12.2 ADT *445*
12.3 ADT Design in C *446*
12.4 Designing your own ADT *446*
Review Questions *448*
Programming Problems *448*

13 DATE—*TODAY WAS TOMORROW!!* **449**

Introduction *449*
13.1 How to Find the Day of Week (Sun, Mon, etc) *453*
13.2 How to Find the Date of the Next N^{th} Sunday, from a given Date *456*
13.3 How to Find the Date of the Previous N^{th} Sunday, from a given Date *458*
13.4 Wrapper Functions: Increases the Readability of Your Code *459*
13.5 Interaction with the System Built in `Date` Structure *464*
13.6 Interaction with the Real World *465*
Review Questions *467*
Programming Problems *468*

14 MAP—*PHONEBOOK, DICTIONARY, CRYPTOGRAPHY* **469**

Introduction *469*
14.1 How to Represent a Map *469*
14.2 How to Define a Predicator over a Map and use it from a Client *474*
14.3 How to know who is who's Friend from the Buddy List *474*
14.4 How to Design a Random Cipher Encoder using a Map *475*
14.5 Application of Map of Maps *476*
14.6 Multilanguage Word Map *477*
14.7 Key Interlinked Map (KIM) *477*
Review Questions *478*
Programming Problems *478*

15 CURRENCY—*NO PRIMITIVE PLEASE!* **479**

Introduction *479*
15.1 A Practical Application: Getting the Lowest Bid Amount *484*

15.2 How to Convert USD to GBP and vice versa *485*
15.3 How to Convert USD to GBP and vice versa Datewise *486*
Review Questions *487*
Programming Problems *487*

16 FILE HANDLING—*SEED, SAVE, SHARE* **488**

Introduction *488*
16.1 What is File? *488*
16.2 What does the Function `rewind`() Do *492*
16.3 How to Simulate UNIX `cat` Command *496*
16.4 How to Simulate UNIX `grep` Command with Exact Match *497*
16.5 How to Simulate UNIX `Grep` Command for Switch–V *498*
16.6 How to Print those Lines of a File that Contain a Word that Sounds like a given Word *503*
16.7 How to Replace a Character in a File with Another Character *503*
16.8 How to Replace a Word in a File with Another Word *504*
16.9 How to Compare Two Text Files Line by Line *505*
16.10 How to Print Same Lines of Two Files *506*
16.11 How to Copy a File from a Source to a Destination *507*
16.12 File Handling in Console-Based Games *524*
16.13 Function Definitions *532*
Review Questions *537*
Programming Problems *538*

Appendix A: Project Ideas! ***539***

Appendix B: Bibliography ***543***

Index ***545***