

Data Structures  
and  
Program Design using



Amiya Kumar Rath  
Alok Kumar Jagadev

SCITECH

*Knowledge for Ages*

ISBN: 978-81-8371-519-5

# DATA STRUCTURES AND PROGRAM DESIGN USING C

Dr. Amiya Kumar Rath

Principal

Dhaneswar Rath Institute of Engineering & Management Studies

Tangi, Cuttack,

Bhubaneswar

Alok Kumar Jagadev

Associate Professor

School of Computer Science & Engineering

ITER, Siksha 'O' Anusandhan University

Bhubaneswar



Publishing for future

**SCITECH PUBLICATIONS (INDIA) PVT. LTD.**

[www.scitechpublications.com](http://www.scitechpublications.com)

[www.scitechpublications.com](http://www.scitechpublications.com)

## CONTENTS

<b>1.</b>	<b>Introduction to Data Structures .....</b>	<b>1.1</b>
<b>1.1</b>	<b>Introduction .....</b>	<b>1.1</b>
<b>1.2</b>	<b>Data Types .....</b>	<b>1.2</b>
	Primitive Data Types .....	1.2
	Integer (int) .....	1.2
	Character (char) .....	1.2
	Floating point Numbers (float) .....	1.2
<b>1.3</b>	<b>Abstract Data Types (ADT) .....</b>	<b>1.3</b>
	Example of an ADT .....	1.4
<b>1.4</b>	<b>Classification of Data Structure .....</b>	<b>1.4</b>
<b>1.5</b>	<b>Description of Some of the Data Structures .....</b>	<b>1.5</b>
1.5.1	Array .....	1.5
1.5.2	List .....	1.6
1.5.3	Stack .....	1.6
1.5.4	Queue .....	1.6
1.5.5	Tree .....	1.7
1.5.6	Graph .....	1.7
<b>1.6</b>	<b>Data Structure Operations .....</b>	<b>1.8</b>
<b>1.7</b>	<b>Introduction to Algorithm .....</b>	<b>1.8</b>
<b>1.8</b>	<b>Algorithm Design and Data Structure .....</b>	<b>1.8</b>
<b>1.9</b>	<b>Algorithm Analysis .....</b>	<b>1.9</b>
<b>1.10</b>	<b>Different Approaches to Designing an Algorithm .....</b>	<b>1.10</b>
	Top-Down Approach .....	1.10
	Bottom-Up Approach .....	1.10
<b>1.11</b>	<b>Time and Space Complexity .....</b>	<b>1.11</b>
	Time Complexity .....	1.11
	Space Complexity .....	1.13
<b>1.12</b>	<b>Asymptotic Notations .....</b>	<b>1.13</b>
1.12.1	O(Big-Oh) Notation .....	1.14
1.12.2	$\Omega$ (Omega) Notation .....	1.15
1.12.3	$\theta$ (Theta)-Notation .....	1.16
1.12.4	Growth of functions .....	1.17
<b>1.13</b>	<b>Algorithm Analysis .....</b>	<b>1.19</b>
	Best-case time complexity .....	1.20
	Worst-case time complexity .....	1.20
	Average-case time complexity .....	1.20
	<b>Summary .....</b>	<b>1.20</b>
	<b>Exercises .....</b>	<b>1.21</b>

<b>2.</b>	<b>Basics Concepts .....</b>	<b>2.1</b>
2.1	<b>Introduction .....</b>	<b>2.1</b>
	Structure of C Program .....	2.2
2.2	<b>Data Types .....</b>	<b>2.2</b>
2.3	<b>Variables .....</b>	<b>2.3</b>
2.4	<b>Constants .....</b>	<b>2.4</b>
2.5	<b>Expressions .....</b>	<b>2.4</b>
2.6	<b>Operators .....</b>	<b>2.4</b>
	Arithmetic Operators .....	2.5
	Unary Operators .....	2.6
	Relational Operators .....	2.7
	Logical Operators .....	2.7
	Assignment Operators .....	2.8
	Conditional Operators .....	2.9
	Bitwise Operators .....	2.9
	Special Operators .....	2.11
	Comma Operator .....	2.11
	sizeof Operator .....	2.11
2.7	<b>Type Casting .....</b>	<b>2.12</b>
2.8	<b>Precedence of Operators .....</b>	<b>2.13</b>
2.9	<b>Control Statements .....</b>	<b>2.13</b>
2.9.1	Branching .....	2.14
	if statement .....	2.14
	if-else statement .....	2.14
	switch statement .....	2.15
2.9.2	goto statement .....	2.16
2.9.3	Loop .....	2.17
	for statement .....	2.17
	while statement .....	2.18
	do-while statement .....	2.18
2.10	<b>Functions .....</b>	<b>2.19</b>
2.10.1	Parameter passing .....	2.22
	Call by value .....	2.22
	Call by reference .....	2.23
2.10.2	Passing arrays to functions .....	2.24
	<b>Summary .....</b>	<b>2.25</b>
	<b>Exercises .....</b>	<b>2.26</b>
<b>3.</b>	<b>Arrays &amp; Strings .....</b>	<b>3.1</b>
3.1	<b>Introduction .....</b>	<b>3.1</b>
3.2	<b>One Dimensional Array .....</b>	<b>3.2</b>
3.2.1	Address calculation .....	3.2

3.2.2	Basic operations of array .....	3.2
	Traversing .....	3.2
	Insertion .....	3.3
	Deletion .....	3.3
	Searching .....	3.4
	Sorting .....	3.4
	Merging .....	3.4
<b>3.3</b>	<b>Two Dimensional Array .....</b>	<b>3.5</b>
3.3.1	Address calculation in two dimensional array .....	3.6
	Row-major order .....	3.6
	Column-major order .....	3.7
<b>3.4</b>	<b>Character Arrays and Strings .....</b>	<b>3.15</b>
3.4.1	Declaration of strings .....	3.16
<b>3.5</b>	<b>String Handling Functions .....</b>	<b>3.17</b>
	Finding the length of a string .....	3.17
	String concatenation .....	3.18
	String reverse .....	3.19
	String copy .....	3.20
	String comparison .....	3.21
	Extract a substring .....	3.23
<b>3.6</b>	<b>String Handling Library Functions .....</b>	<b>3.24</b>
	strcpy() .....	3.24
	strcat() .....	3.25
	strlen() .....	3.25
	strcmp() .....	3.25
	strlwr() .....	3.25
	strupr() .....	3.26
<b>3.7</b>	<b>Array of Strings .....</b>	<b>3.26</b>
<b>3.8</b>	<b>Pattern Matching .....</b>	<b>3.28</b>
	Summary .....	3.30
	Exercises .....	3.30
<b>4.</b>	<b>Pointers and Structures</b>	
4.1	Introduction .....	4.1
4.2	Pointer Variable Declarations .....	4.2
4.3	Pointers and Arrays .....	4.3
4.4	Array of Pointers .....	4.6
4.5	Pointers and Strings .....	4.7
4.6	Functions with Variable Number of Arguments .....	4.8
4.7	Dynamic Memory Allocation .....	4.10
4.8	Multiple Indirection .....	4.12
4.9	Structures .....	4.13

4.10	Passing Structures to Functions .....	4.17
4.11	Enumerations .....	4.20
4.12	User-defined data type (typedef) .....	4.22
	Summary .....	4.23
	Exercises .....	4.23
<b>5.</b>	<b>Single Linked List .....</b>	<b>5.1</b>
5.1	Introduction .....	5.1
5.2	Single Linked List .....	5.2
5.3	Static Representation of Linked List .....	5.2
	Static implementation of linked list .....	5.2
5.4	Overview of Dynamic Memory Allocation .....	5.9
	Allocation of memory .....	5.9
	Releasing Memory .....	5.10
5.5	Dynamic Representation of Linked List .....	5.10
5.6	Operations in a Single Linked List .....	5.11
5.6.1	Traversing a single linked list .....	5.11
5.6.2	Insertion of a node in a single linked list .....	5.14
	Insert at beginning .....	5.14
	Insert at end .....	5.15
	Insert at specific location .....	5.16
	Insert a node after a given node .....	5.16
5.6.3	Deletion of a node from a single linked list .....	5.17
	Delete the first node .....	5.18
	Delete of last node .....	5.18
	Delete of a node at specific location .....	5.19
	Delete after a given node .....	5.20
	Searching in a single linked list .....	5.30
	Sorting in a single linked list .....	5.34
	Reversing a single linked list .....	5.38
	Merging of two linked lists into a larger list .....	5.43
5.7	Circular Single Linked List .....	5.46
5.8	Header Linked List .....	5.49
5.8.1	Single linked list with header node .....	5.49
5.8.2	Circular single linked list with header node .....	5.52
5.9	Linked Lists versus Arrays .....	5.55
	Summary .....	5.55
	Exercises .....	5.56
<b>6.</b>	<b>Double Linked List .....</b>	<b>6.1</b>
6.1	Introduction .....	6.1
6.2	Operations in a Double Linked List .....	6.2

6.2.1	Traversing a double linked list .....	6.2
6.2.2	Insertion of a node in a double linked list .....	6.6
	Insert at beginning .....	6.6
	Insert at end .....	6.7
	Insert at specific location .....	6.8
	Insert a node after a given node .....	6.8
6.2.3	Deletion of a node from a double linked list .....	6.9
	Delete the first node .....	6.10
	Deletion of last node .....	6.10
	Deletion of a node at specific location .....	6.11
	Deletion after a given node .....	6.12
<b>6.3</b>	<b>Circular Double Linked List .....</b>	<b>6.23</b>
<b>6.4</b>	<b>Header Linked List .....</b>	<b>6.26</b>
6.4.1	Double linked list with header node .....	6.26
6.4.2	Circular double linked list with header node .....	6.29
	<b>Summary .....</b>	<b>6.32</b>
	<b>Exercises .....</b>	<b>6.33</b>
<b>7.</b>	<b>Sparse Matrix &amp; Polynomials .....</b>	<b>7.1</b>
7.1	<b>Introduction .....</b>	<b>7.1</b>
7.2	<b>Sparse Matrices .....</b>	<b>7.2</b>
	7.2.1 Array representation of sparse matrix .....	7.2
	7.2.2 Linked representation of sparse matrix .....	7.2
7.3	<b>Operations on Sparse Matrix .....</b>	<b>7.3</b>
	7.3.1 Transpose of a sparse matrix .....	7.3
	7.3.2 Fast transpose .....	7.6
	7.3.3 Addition of two sparse matrices .....	7.9
7.4	<b>Polynomials .....</b>	<b>7.18</b>
	7.4.1 Array representation of polynomial .....	7.18
	7.4.2 Linked representation of polynomial .....	7.22
	<b>Summary .....</b>	<b>7.27</b>
	<b>Exercises .....</b>	<b>7.27</b>
<b>8.</b>	<b>Stacks and Recursion .....</b>	<b>8.1</b>
8.1	<b>Introduction .....</b>	<b>8.1</b>
8.2	<b>Stacks .....</b>	<b>8.2</b>
	8.2.1 Stack Terminologies .....	8.2
	8.2.2 Stack operations .....	8.3
8.3	<b>Representation of Stack .....</b>	<b>8.3</b>
	8.3.1 Array representation of stack .....	8.3
	8.3.2 Linked list representation of stack .....	8.9
8.4	<b>Applications of Stack .....</b>	<b>8.14</b>
	8.4.1 Evaluation of arithmetic expression .....	8.14

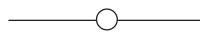
8.4.2	Infix, prefix and postfix notations .....	8.16
	Infix notation .....	8.16
	Prefix notation .....	8.16
	Postfix notation .....	8.16
8.4.3	Conversion of expressions .....	8.17
8.4.4	Infix to postfix expression .....	8.19
8.4.5	Evaluation of postfix expression .....	8.25
8.4.6	Infix to prefix conversion .....	8.29
8.4.7	Evaluation of prefix expression .....	8.31
8.4.8	Recursion .....	8.32
	Factorial of a number .....	8.33
	Fibonacci Series .....	8.35
	Binary Search .....	8.37
	Ackermann Function .....	8.39
	Towers of Hanoi .....	8.41
	<b>Summary .....</b>	<b>8.44</b>
	<b>Exercises .....</b>	<b>8.44</b>
<b>9.</b>	<b>Queues .....</b>	<b>9.1</b>
9.1	<b>Introduction .....</b>	<b>9.1</b>
9.2	<b>Queues .....</b>	<b>9.2</b>
	9.2.1 Queue terminologies .....	9.2
	9.2.2 Queue operations .....	9.3
9.3	<b>Representation of Queue .....</b>	<b>9.3</b>
	9.3.1 Array representation of queue .....	9.3
	9.3.2 Limitation of queue in array representation .....	9.9
	9.3.3 Circular queue .....	9.10
	9.3.4 Array representation of circular queue .....	9.11
	Insertion .....	9.12
	Deletion .....	9.13
	9.3.5 Linked list representation of a queue .....	9.18
9.4	<b>Double Ended Queue (Deque) .....</b>	<b>9.27</b>
9.5	<b>Priority Queue .....</b>	<b>9.35</b>
9.6	<b>Application of Queue .....</b>	<b>9.36</b>
	9.6.1 Simulation .....	9.36
	Simulation of a traffic control system .....	9.37
	9.6.2 CPU scheduling in multiprogramming and time sharing environment .....	9.38
	Round Robin scheduling algorithm .....	9.38
	<b>Summary .....</b>	<b>9.39</b>
	<b>Exercises .....</b>	<b>9.39</b>
<b>10.</b>	<b>Trees .....</b>	<b>10.1</b>
10.1	<b>Introduction .....</b>	<b>10.1</b>



<b>10.2</b>	<b>Definition .....</b>	<b>10.2</b>
	Application of Trees .....	10.2
<b>10.3</b>	<b>Tree Terminology .....</b>	<b>10.2</b>
<b>10.4</b>	<b>Binary Tree .....</b>	<b>10.3</b>
	10.4.1 Properties of binary tree .....	10.4
	10.4.2 Strictly binary tree .....	10.4
	10.4.3 Complete binary tree .....	10.4
	10.4.4 Almost complete binary tree .....	10.5
<b>10.5</b>	<b>Memory Representation of Binary Tree .....</b>	<b>10.5</b>
	10.5.1 Array representation of binary tree .....	10.5
	10.5.2 Linked representation of binary tree .....	10.6
<b>10.6</b>	<b>Binary Tree Traversal .....</b>	<b>10.6</b>
	Preorder traversal .....	10.6
	Inorder traversal .....	10.7
	Postorder traversal .....	10.7
	10.6.1 Recursive algorithm for tree traversal .....	10.7
	10.6.2 Nonrecursive algorithm for tree traversal .....	10.8
<b>10.7</b>	<b>Construction of Binary Tree .....</b>	<b>10.13</b>
<b>10.8</b>	<b>Expression Trees .....</b>	<b>10.15</b>
	10.8.1 Construction of expression tree .....	10.15
	Construction of a tree from a postfix expression .....	10.15
	Construction of a tree from a prefix expression .....	10.17
<b>10.9</b>	<b>Binary Search Trees (BST) .....</b>	<b>10.19</b>
	10.9.1 Operations on a binary search tree .....	10.20
	Searching a node in BST .....	10.20
	Insertion of a node in BST .....	10.20
	Deletion of a node in BST .....	10.21
	10.9.2 Efficiency of BST operations .....	10.23
<b>10.10</b>	<b>Threaded Binary Tree .....</b>	<b>10.41</b>
<b>10.11</b>	<b>Height Balanced Trees: AVL Tree .....</b>	<b>10.45</b>
	10.11.1 Insertion of a node into an AVL tree .....	10.46
	10.11.2 AVL rotations .....	10.47
<b>10.12</b>	<b>General Tree .....</b>	<b>10.57</b>
	10.12.1 Memory representation of general trees .....	10.57
	10.12.2 Forest .....	10.58
	10.12.3 Binary tree representation of general tree .....	10.58
	10.12.4 Binary tree representation of a forest .....	10.59
<b>10.13</b>	<b>Multiway Search Tree .....</b>	<b>10.59</b>
	Balanced multiway search tree .....	10.59
	10.13.1 B-Tree .....	10.60
	Insertion in B-tree .....	10.60
	Deletion in B-tree .....	10.61
<b>10.14</b>	<b>B+ Tree .....</b>	<b>10.63</b>

Summary .....	10.64
Exercises .....	10.65
<b>11. Graphs .....</b>	<b>11.1</b>
<b>11.1 Introduction .....</b>	<b>11.1</b>
<b>11.2 Terminologies of Graph .....</b>	<b>11.2</b>
<b>11.3 Representation of Graph .....</b>	<b>11.5</b>
11.3.1 Sequential representation of graph .....	11.5
Adjacency matrix .....	11.5
Incidence Matrix .....	11.7
Path matrix .....	11.8
11.3.2 Linked representation of graph .....	11.8
<b>11.4 Traversal in Graph .....</b>	<b>11.13</b>
11.4.1 Depth first search .....	11.13
11.4.2 Breadth first search .....	11.20
<b>11.5 Shortest Path Algorithm .....</b>	<b>11.26</b>
11.5.1 Warshall's Algorithm .....	11.26
11.5.2 Modified Warshall's Algorithm .....	11.32
11.5.3 Single-source shortest path .....	11.38
Dijkstra's Algorithm .....	11.38
<b>11.6 Minimum Spanning Trees .....</b>	<b>11.44</b>
11.6.1 Kruskal's Algorithm .....	11.45
11.6.2 Prim's Algorithm .....	11.47
<b>11.7 Topological Sorting .....</b>	<b>11.48</b>
Summary .....	11.52
Exercises .....	11.54
<b>12. Searching &amp; Sorting .....</b>	<b>12.1</b>
<b>12.1 Introduction .....</b>	<b>12.1</b>
<b>12.2 Types of Searching Techniques .....</b>	<b>12.2</b>
12.2.1 Linear search .....	12.2
12.2.2 Binary search .....	12.5
<b>12.3 Sorting .....</b>	<b>12.9</b>
12.3.1 Bubble sort .....	12.9
12.3.2 Insertion sort .....	12.12
12.3.3 Selection Sort .....	12.14
12.3.4 Radix sort .....	12.17
12.3.5 Shell sort .....	12.20
Gap Sequence .....	12.21
12.3.6 Divide-and-conquer approach .....	12.24
12.3.7 Merge sort .....	12.24
12.3.8 Quick sort .....	12.28
12.3.9 Heap sort .....	12.35

12.3.10	Address calculation sort .....	12.39
<b>12.4</b>	<b>Comparison of Algorithms .....</b>	<b>12.43</b>
	<b>Summary .....</b>	<b>12.43</b>
	<b>Exercises .....</b>	<b>12.44</b>
<b>13.</b>	<b>Hashing .....</b>	<b>13.1</b>
<b>13.1</b>	<b>Introduction .....</b>	<b>13.1</b>
<b>13.2</b>	<b>Hash Functions .....</b>	<b>13.2</b>
	Division method .....	13.2
	Mid-Square method .....	13.3
	Folding method .....	13.3
	Digit analysis method .....	13.4
<b>13.3</b>	<b>Collision Resolution Techniques .....</b>	<b>13.4</b>
13.3.1	Closed hashing (Open addressing) .....	13.5
	Linear probing .....	13.5
	Quadratic Probing .....	13.6
	Random probing .....	13.7
	Double hashing .....	13.7
	Rehashing .....	13.8
13.3.2	Open hashing (chaining) .....	13.8
	<b>Summary .....</b>	<b>13.9</b>
	<b>Exercises .....</b>	<b>13.10</b>
<b>14.</b>	<b>Storage Management .....</b>	<b>14.1</b>
<b>14.1</b>	<b>Introduction .....</b>	<b>14.1</b>
<b>14.2</b>	<b>Dynamic Memory Management .....</b>	<b>14.2</b>
14.2.1	First fit method .....	14.2
14.2.2	Best fit method .....	14.4
14.2.3	Comparison between first fit and best fit methods .....	14.5
14.2.4	Worst fit Method .....	14.5
<b>14.3</b>	<b>Boundary Tag System .....</b>	<b>14.6</b>
<b>14.4</b>	<b>Buddy System .....</b>	<b>14.7</b>
<b>14.5</b>	<b>Compaction .....</b>	<b>14.10</b>
<b>14.6</b>	<b>Garbage Collection .....</b>	<b>14.11</b>
	<b>Summary .....</b>	<b>14.11</b>
	<b>Exercises .....</b>	<b>14.12</b>

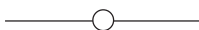


## LIST OF PROGRAMS

Program 2.1:	Sum of two numbers .....	2.3
Program 2.2:	Illustration of pre and post increment and decrement operators .....	2.6
Program 2.3:	Convert centigrade to Fahrenheit and vice versa .....	2.8
Program 2.4:	Find the larger of two numbers using conditional operator .....	2.9
Program 2.5:	Illustration of bitwise operators .....	2.11
Program 2.6:	Determine the size of different types of data .....	2.11
Program 2.7:	Illustration of type casting .....	2.12
Program 2.8:	Display absolute value of an input number .....	2.14
Program 2.9:	Find the larger of two numbers .....	2.15
Program 2.10:	Illustration of switch case statement .....	2.15
Program 2.11:	Illustration of goto statement .....	2.16
Program 2.12:	Find the sum of given range of numbers .....	2.17
Program 2.13:	Sum of digits of a given number .....	2.18
Program 2.14:	Check a given number is palindrome or not .....	2.19
Program 2.15:	Sum of two numbers using function .....	2.20
Program 2.16:	Implementation of call by value parameter passing mechanism .....	2.22
Program 2.17:	Implementation of call by address parameter passing mechanism .....	2.23
Program 2.18:	Find the average of n numbers .....	2.24
Program 3.1:	Addition of two matrices .....	3.7
Program 3.2:	Subtraction of two matrices .....	3.9
Program 3.3:	Multiplication of two matrices .....	3.11
Program 3.4:	Find out the sum of each row and column of a matrix .....	3.12
Program 3.5:	Sum of diagonal element of a matrix .....	3.13
Program 3.6:	Transpose of a matrix .....	3.14
Program 3.7:	Use of gets() function .....	3.16
Program 3.8:	Use of putchar() function .....	3.16
Program 3.9:	Use of puts() function .....	3.17
Program 3.10:	Find the length of the string without using strlen() function .....	3.17
Program 3.11:	Find the length of the string using strlen() function .....	3.18
Program 3.12:	String concatenation without using the function strcat() .....	3.18
Program 3.13:	String concatenation using strcat() function .....	3.19
Program 3.14:	Reverse a string .....	3.19
Program 3.15:	Check whether the given string is palindrome or not .....	3.20
Program 3.16:	String copy without using strcpy() function .....	3.20
Program 3.17:	String copy using strcpy() function .....	3.21
Program 3.18:	String comparison without using strcmp() function .....	3.21
Program 3.19:	Comparison of two strings .....	3.22

Program 3.20:	Find a substring from a given string .....	3.23
Program 3.21:	Converts string into lower and upper case .....	3.26
Program 3.22:	Sort array of strings .....	3.27
Program 3.23:	Search for a pattern in a string .....	3.29
Program 4.1:	Representation of array elements .....	4.4
Program 4.2:	Accessing two-dimensional array elements .....	4.6
Program 4.3:	Illustration of array of pointers .....	4.6
Program 4.4:	Illustration of pointers to string .....	4.7
Program 4.5:	A function to sum a variable number of integers .....	4.10
Program 4.6:	Illustration of dynamic memory allocation .....	4.11
Program 4.7:	Illustration of pointer to pointer .....	4.13
Program 4.8:	Illustration of structure .....	4.14
Program 4.9:	Illustration of pointer to structure .....	4.16
Program 4.10:	Illustration of pointer to the structure as an argument to a function .....	4.19
Program 4.11:	Illustration of enum .....	4.21
Program 4.12:	Given character is a operator or operand .....	4.21
Program 5.1:	Static implementation of linked list .....	5.4
Program 5.2:	Creating a single linked list .....	5.11
Program 5.3:	Implementation of single linked list operations .....	5.20
Program 5.4:	Searching an element from a single linked list .....	5.30
Program 5.5:	Sort the elements of a single linked list .....	5.35
Program 5.6:	Reversing a single linked list .....	5.39
Program 5.7:	Merging of two lists into a single list .....	5.44
Program 5.8:	Creating a circular single linked list .....	5.47
Program 5.9:	Creating a single linked list with header node .....	5.50
Program 5.10:	Creating a circular single linked list with header node .....	5.52
Program 6.1:	Creating a double linked list .....	6.3
Program 6.2:	Implementation of double linked list operations .....	6.13
Program 6.3:	Creating a circular double linked list .....	6.24
Program 6.4:	Creating a header double linked list .....	6.27
Program 6.5:	Creating a circular double linked list with header node .....	6.29
Program 7.1:	Finding of transpose of a sparse matrix using array .....	7.4
Program 7.2:	Transpose of a sparse matrix using faster technique .....	7.7
Program 7.3:	Addition of two sparse matrices .....	7.9
Program 7.4:	Addition of two sparse matrices using linked list .....	7.13
Program 7.5:	Operations of polynomials using array .....	7.18
Program 7.6:	Operations of polynomials using linked list .....	7.22
Program 8.1:	Array representation of stack .....	8.5
Program 8.2:	Linked list representation of stack .....	8.10
Program 8.3:	Infix to postfix conversion .....	8.20

Program 8.4:	Evaluation of postfix expression .....	8.27
Program 8.5:	Factorial of a number using recursion .....	8.35
Program 8.6:	Fibonacci series using recursion .....	8.36
Program 8.7:	Fibonacci series using non recursion .....	8.36
Program 8.8:	Implementation of binary search .....	8.38
Program 8.9:	Implementation of ackermann function .....	8.40
Program 8.10:	Implementation of towers of hanoi .....	8.42
Program 9.1:	Array representation of queue .....	9.5
Program 9.2:	Array representation of circular queue .....	9.14
Program 9.3:	Implementing queue using linked list .....	9.19
Program 9.4:	Linked list implementation of queue using two structures .....	9.23
Program 9.5:	Implementing deque using linked list .....	9.30
Program 10.1:	Implementation of BST using array .....	10.23
Program 10.2:	Implementation of BST using linked list .....	10.28
Program 10.3:	Implementation of non recursive inorder, preorder and postorder traversals in C .....	10.32
Program 10.4:	Creation of BST and different traversal operations using recursion .....	10.37
Program 10.5:	Creation of threaded BST .....	10.42
Program 10.6:	Implementation of AVL tree .....	10.52
Program 11.1:	Implementation of adjacency list representation of graph .....	11.10
Program 11.2:	Implementation of graph traversal using depth first search .....	11.15
Program 11.3:	Implementation of graph traversal using breadth first search .....	11.21
Program 11.4:	Implementation of Warshall's algorithm to find path matrix .....	11.29
Program 11.5:	Implementation of revised Warshall's Algorithm to find shortest path matrix .....	11.35
Program 11.6:	Find shortest path using Dijkstra's algorithm .....	11.41
Program 12.1:	Implementation of linear search .....	12.4
Program 12.2:	Implementation of binary search .....	12.7
Program 12.3:	Implementation of bubble sort .....	12.11
Program 12.4:	Implementation of Insertion sort .....	12.13
Program 12.5:	Implementation of selection sort .....	12.16
Program 12.6:	Implementation of radix sort .....	12.19
Program 12.7:	Implementation of shell sort .....	12.23
Program 12.8:	Implementation of merge sort .....	12.27
Program 12.9:	Implementation of quick sort .....	12.33
Program 12.10:	Implementation of heap sort .....	12.38
Program 12.11:	Implementation of address calculation sort .....	12.40



# Data Structures and Program Design using C

The text emphasizes details about the concept of different data structures and their implementation. The level of presentation is kept simple and illustrative so that even average readers can get maximum benefit out of it. Practically this book will provide everything the reader need to know on data structure.

## Salient features

- ★ Follows the development of data structure from its theoretical conception to its concrete realization.
- ★ Provides a comprehensive coverage of the subject.
- ★ Demonstrates the development of algorithms in a simpler manner.
- ★ Demonstrates the implementation of algorithms in a good programming style.
- ★ Includes numerous illustrative examples.

**Amiya Kumar Rath** obtained his B.E. Degree in Computer Science & Engineering from Marathwada University (presently known as Dr. Babasaheb Ambedkar Marathwada University), Maharashtra in the year 1990. Subsequently, acquired MBA degree in systems Management from Shivaji University in the year 1993 and M.Tech in Computer Science from Utkal University in the year 2001. The same University awarded him with the Ph.D degree for his work in the field of Embedded system. He has served in various positions in different premier Institutes. Presently he is with DRIEMS, Tangi, Cuttack as Principal. Dr.Rath has contributed more than 50 papers to many journals and proceedings. He has also contributed five books. His research interest includes Embedded System, Adhoc Networks, Evolutionary Computation and Data Mining.

Dr. Rath can be reached at [www.amiyakumarrath.com](http://www.amiyakumarrath.com)

**Alok Kumar Jagadev** obtained his MCA degree from Regional Institute of Technology (Presently knows as National Institute of Technology), Jamshedpur in the year 1992. Subsequently, acquired M.Tech in Computer Science from Utkal University in the year 2001. He obtained his Ph.D degree for his work in the field of Wireless Adhoc Networks from Siksha 'O' Anusandhan University, Bhubaneswar. He has served in various positions in different premier Institutes. Presently he is with ITER, Siksha 'O' Anusandhan University, Bhubaneswar as Associate Professor in the School of Computer Science & Engineering. He has contributed more than 30 papers to many journals and proceedings. He has also contributed four books. His research interest includes Adhoc Networks and Application of Soft Computing Techniques in the different areas of data mining.



Publishing  
for  
future

**SCITECH PUBLICATIONS (INDIA) PVT. LTD.**

[www.scitechpublications.com](http://www.scitechpublications.com)

email: [scitechcorp@yahoo.co.in](mailto:scitechcorp@yahoo.co.in)



Buy books online @  
[www.scitechpublications.com](http://www.scitechpublications.com)

