



Contents

Acknowledgments	xiii
Preface	xv
Part I: C Language	I
1 Introduction to the C Language	3
The first program in C	3
Inputting the data	5
The control statement (if statement)	6
The iteration loop (for loop)	7
The do...while loop	9
The switch statement	10
2 Data Types	13
Various data types in C	14
The integer data type family	16
Overflow in char and unsigned char data types	18
The char type	19
Octal numbers	21
Hexadecimal numbers	21
Representation of floating-point numbers	21
Type conversion	25
Forced conversion	26
Type casting	27
3 C Operators	31
Arithmetic operator	32
Relational operator	34
Logical operator	35
Ternary operator	37
Increment operator	38
Comma operator	39
Bitwise operator	40
Operator precedence	43

vi C & Data Structures

4	Control Structures	45
	Control structures	45
	The if statement	46
	Scope of an if clause	47
	The if-else statement	47
	The if-else if statement	48
	The switch statement	50
	The while loop	51
	The do-while loop	52
	The for loop	53
	The for loop with a comma operator	54
	The break statement	54
	The continue statement	55
5	The printf Function	57
	printf	57
	Placeholders	58
6	Address and Pointers	63
	Address	63
	Pointers	64
7	The scanf Function	67
	scanf	67
	The scanf placeholders	68
8	Preprocessing	71
	Preprocessor	71
	undef	72
	ifdef	73
	ifndef	75
	#if	76
	ifndef	77
	ifelif	78
	Error directive	80
	#line	81
	Macro	82
	Macro and function	83

9 Arrays	85
Arrays	85
Address of each element in an array	86
Accessing an array using pointers	88
Manipulating arrays using pointers	89
Another case of manipulating an array using pointers	91
Two-dimensional array	93
Three-dimensional array	95
Pointer arrays	97
 10 Functions	 101
Functions	101
The concept of stack	103
The sequence of execution during a function call	104
Parameter passing	106
Call by reference	107
The concept of global variables	108
Resolving variable references	109
Syntax of function definition	111
Calling function	113
 11 Storage of Variables	 115
Storage	115
External references	117
Register variables	118
Scope of variables	120
Further scope of variables	121
 12 Memory Allocation	 123
Dynamic memory allocations	123
 13 Recursion	 127
Recursion	127
Stack overheads in recursion	128
Writing a recursive function	130
 14 Strings	 133
Strings as an array of characters	133
String definition	134
Strings as parameters	135

viii C & Data Structures

15 Structures	137
Structures	137
Complex structure definitions	138
Memory allocation to structure	140
Programming with structures	141
Structure pointers	143
16 Union	145
Union	145
17 Files	147
The concept of files	147
Direct access files	156

Part II: Data Structures 165

18 Arrays, Searching, and Sorting	167
Arrays	167
Application of arrays	170
Manipulations on the list implemented using an array	173
Merging of two sorted lists	180
Transpose of a matrix	184
Finding the saddle point of a matrix	189
Implementation of heaps	192
Sorting and searching	192
Bubble sort	193
Quick sort	195
Merge sort	201
Heapsort	206
Searching techniques: linear or sequential search	210
Binary search	213
Hashing	217
Hashing functions	220
19 Stacks and Queues	227
The concept of stacks and queues	227
Stacks	227
Applications of stacks	235

Queues	242
Implementation of queues	243
Circular queues	247
Implementation of a queue using linked representation	251
Applications of queues	256
20 Linked Lists	263
The concept of the linked lists	263
Inserting a node by using recursive programs	267
Sorting and reversing a linked list	269
Deleting the specified node in a singly linked list	276
Inserting a node after the specified node in a singly linked list	280
Inserting a new node in a sorted list	284
Counting the number of nodes of a linked list	289
Merging of two sorted lists	291
Erasing a linked list	299
Polynomial representation	303
Representation of sparse matrices	308
Circular linked lists	315
Splitting a list with $2n$ nodes into two separate and equal lists	318
Merging of two circular lists	321
Reversing the direction of links in a singly linked circular list	325
Doubly linked lists	330
Insertion of a node in a doubly linked list	335
Deleting a node from a doubly linked list	339
Application of doubly linked lists to memory management	343
21 Trees	347
The concept of tree	347
Binary tree and its representation	349
Binary tree traversal	354
Binary search tree	357
Counting the number of nodes in a binary search tree	364
Swapping of left and right subtrees of a given binary tree	367
Searching for a target key in a binary search tree	370
Deletion of a node from binary search tree	375

x C & Data Structures

22 Graphs	387
Graphs	387
Representations of a graph	390
Computing indegree and outdegree of a node of a graph using adjacency matrix representation	393
Depth-first traversal	396
Breadth-first traversal	399
Connected component of a graph	403
Depth-first spanning tree and breadth-first spanning tree	410
Minimum-cost spanning tree	413
Directed acyclic graph (DAG)	420

Part III: Advanced Problems in Data Structures 431

23 Problems in Arrays, Searching, Sorting, Hashing	433
Problem: Calculate the value of an $N \times N$ determinant	433
Problem: Write a program to find the saddle point of a matrix, if it exists	439
Problem: Multiply two sparse matrices	441
Problem: Multiplication of two sparse matrices (different versions)	448
Problem: Implement K -way sort-merge to sort a file containing records	455
Problem: Find a plateau in a matrix	462
Problem: Implementation of a hash search	466
Problem: Implementation of rehashing	470
24 Problems in Stacks and Queues	477
Problem: Convert an infix expression to prefix form	477
Problem: Implementation of two stacks using an array	483
25 Problems in Linked Lists	487
Problem: Implementation of polynomials using linked lists	487
Problem: Implementation of circular lists by using arrays	493
Problem: Reversing links in the case of circular list	498
Problem: Memory management using lists	502
Problem: Memory management using various schemes	505
Problem: Garbage collection—the first method	513
Problem: Garbage collection—the second method	520
Problem: Compute N equivalence classes	528

26 Problems in Strings	533
Problem: Maximize a combination under constraints	533
Problem: Maximize a combination of strings—the second method	538
Problem: Closure of sets	542
Problem: Finding the maximum matching pattern in the string	548
Problem: Implementation of the soundex function	550
 27 Problems in Trees	 555
Problem: Write a non-recursive version of preorder	555
Problem: Write a non-recursive version of postorder	561
Problem: Preorder traversal of a threaded binary tree	567
Problem: Implementation of a set using a binary tree	573
Problem: Huffman coding	578
Problem: Implementation of a B-tree	584
Problem: Implementation of a B+ tree	603
 28 Problems in Graphs	 617
Problem: The dfs method for graph traversal	617
Problem: Connected components in a graph	622
Problem: Minimum spanning tree	627
Problem: Topological sort	632
Problem: Finding the shortest path by using an adjacency matrix	637
Problem: Finding the shortest path by using an adjacency list	641
Problem: The m shortest path	648
Problem: The all-cost shortest path	661
 29 Miscellaneous Problems	 667
Problem: The two-class classification problem	667
Problem: The <i>N</i> -coins problem	673
Problem: All combinations of strings	678
Problem: The 8-knights problem	680
Problem: <i>N</i> -queens problem	684
Problem: Mapping of <i>N</i> -queues in an array	687
Problem: Implementation of A* algorithm	692
 Index	 697