



SECOND EDITION

Data Structures Using **C**

A. K. Sharma

ALWAYS LEARNING

PEARSON

Contents

<i>Preface to the Second Edition</i>	<i>xiii</i>
<i>Preface</i>	<i>xiv</i>
<i>About the Author</i>	<i>xv</i>
Chapter 1: Overview of C	1
1.1 The History	1
1.2 Characters Used in C	2
1.3 Data Types	2
1.3.1 Integer Data Type (int)	2
1.3.2 Character Data Type (char)	3
1.3.3 The Floating Point (float) Data Type	3
1.4 C Tokens	4
1.4.1 Identifiers	4
1.4.2 Keywords	5
1.4.3 Variables	5
1.4.4 Constants	7
1.5 Structure of a C Program	8
1.5.1 Our First Program	8
1.6 printf() and scanf() Functions	8
1.6.1 How to Display Data Using printf() Function	9
1.6.2 How to Read Data from Keyboard Using scanf()	10
1.7 Comments	10
1.8 Escape Sequence (Backslash Character Constants)	11
1.9 Operators and Expressions	13
1.9.1 Arithmetic Operators	13
1.9.2 Relational and Logical Operators	14
1.9.3 Conditional Operator	16
1.9.4 Order of Evaluation of Expressions	17
1.9.5 Some Special Operators	18
1.9.6 Assignment Operator	18
1.9.7 Bitwise Shift Operators	19
1.10 Flow of Control	20
1.10.1 The Compound Statement	21
1.10.2 Selective Execution (Conditional Statements)	21
1.10.3 Repetitive Execution (Iterative Statements)	25
1.10.4 The exit() Function	27
1.10.5 Nested Loops	28
1.10.6 The Goto Statement (Unconditional Branching)	28

1.11	Input-Output Functions (I/O)	30
1.11.1	Buffered I/O	31
1.11.2	Single Character Functions	32
1.11.3	String-based Functions	33
1.12	Arrays	34
1.13	Structures	34
1.13.1	Defining a Structure in C	35
1.13.2	Referencing Structure Elements	36
1.13.3	Arrays of Structures	36
1.13.4	Initializing Structures	37
1.13.5	Assignment of Complete Structures	37
1.13.6	Nested Structures	38
1.14	User-defined Data Types	39
1.14.1	Enumerated Data Types	40
1.15	Unions	42
1.16	Functions	43
1.16.1	Function Prototypes	44
1.16.2	Calling a Function	45
1.16.3	Parameter Passing in Functions	47
1.16.4	Returning Values from Functions	52
1.16.5	Passing Structures to Functions	52
1.17	Recursion	56
1.17.1	Types of Recursion	60
1.17.2	Tower of Hanoi	65

Chapter 2: Data Structures and Algorithms: An Introduction 72

2.1	Overview	72
2.2	Concept of Data Structures	73
2.2.1	Choice of Right Data Structures	74
2.2.2	Types of Data Structures	76
2.2.3	Basic Terminology Related with Data Structures	77
2.3	Design of a Suitable Algorithm	78
2.3.1	How to Develop an Algorithm?	78
2.3.2	Stepwise Refinement	80
2.3.3	Using Control Structures	81
2.4	Algorithm Analysis	85
2.4.1	Big-Oh Notation	86

Chapter 3: Arrays: Searching and Sorting 93

3.1	Introduction	93
3.2	One-dimensional Arrays	94
3.2.1	Traversal	95
3.2.2	Selection	96
3.2.3	Searching	98

3.2.4	Insertion and Deletion	105
3.2.5	Sorting	109
3.3	Multi-dimensional Arrays	130
3.4	Representation of Arrays in Physical Memory	134
3.4.1	Physical Address Computation of Elements of One-dimensional Arrays	135
3.4.2	Physical Address Computation of Elements of Two-dimensional Arrays	136
3.5	Applications of Arrays	138
3.5.1	Polynomial Representation and Operations	138
3.5.2	Sparse Matrix Representation	141

Chapter 4: Stacks and Queues 151

4.1	Stacks	151
4.1.1	Stack Operations	152
4.2	Applications of Stacks	156
4.2.1	Arithmetic Expressions	156
4.3	Queues	170
4.3.1	Queue Operations	171
4.3.2	Circular Queue	176
4.3.3	Priority Queue	181
4.3.4	The Deque	185

Chapter 5: Pointers 197

5.1	Introduction	197
5.1.1	The '&' Operator	197
5.1.2	The '*' Operator	198
5.2	Pointer Variables	198
5.2.1	Dangling Pointers	202
5.3	Pointers and Arrays	203
5.4	Array of Pointers	208
5.5	Pointers and Structures	208
5.6	Dynamic Allocation	210
5.6.1	Self Referential Structures	215

Chapter 6: Linked Lists 227

6.1	Introduction	227
6.2	Linked Lists	227
6.3	Operations on Linked Lists	231
6.3.1	Creation of a Linked List	231
6.3.2	Travelling a Linked List	236
6.3.3	Searching a Linked List	241
6.3.4	Insertion in a Linked List	243
6.3.5	Deleting a Node from a Linked List	250

6.4	Variations of Linked Lists	253
6.4.1	Circular Linked Lists	254
6.4.2	Doubly Linked List	258
6.5	The Concept of Dummy Nodes	264
6.6	Linked Stacks	266
6.7	Linked Queues	270
6.8	Comparison of Sequential and Linked Storage	274
6.9	Solved Problems	274

Chapter 7: Trees 282

7.1	Introduction	282
7.2	Basic Terminology	284
7.3	Binary Trees	285
7.3.1	Properties of Binary Trees	286
7.4	Representation of a Binary Tree	287
7.4.1	Linear Representation of a Binary Tree	287
7.4.2	Linked Representation of a Binary Tree	289
7.4.3	Traversal of Binary Trees	291
7.5	Types of Binary Trees	298
7.5.1	Expression Tree	298
7.5.2	Binary Search Tree	303
7.5.3	Heap Trees	319
7.5.4	Threaded Binary Trees	340
7.6	Weighted Binary Trees and Huffman Algorithm	352
7.6.1	Huffman Algorithm	354
7.6.2	Huffman Codes	356
7.7	Dynamic Dictionary Coding	360

Chapter 8: Graphs 365

8.1	Introduction	365
8.2	Graph Terminology	366
8.3	Representation of Graphs	368
8.3.1	Array-based Representation of Graphs	368
8.3.2	Linked Representation of a Graph	371
8.3.3	Set Representation of Graphs	373
8.4	Operations of Graphs	373
8.4.1	Insertion Operation	374
8.4.2	Deletion Operation	379
8.4.3	Traversal of a Graph	384
8.4.4	Spanning Trees	396
8.4.5	Shortest Path Problem	401
8.5	Applications of Graphs	408

Chapter 9: Files	412	
9.1 Data and Information	412	
9.1.1 Data	412	
9.1.2 Information	412	
9.2 File Concepts	413	
9.3 File Organization	415	
9.4 Files in C	416	
9.5 Files and Streams	416	
9.6 Working with Files Using I/O Stream	418	
9.6.1 Opening of a File	418	
9.6.2 Unformatted File I/O Operations	419	
9.6.3 Formatted File I/O Operations	425	
9.6.4 Reading or Writing Blocks of Data in Files	426	
9.7 Sequential File Organization	430	
9.7.1 Creating a Sequential File	430	
9.7.2 Reading and Searching a Sequential File	431	
9.7.3 Appending a Sequential File	431	
9.7.4 Updating a Sequential File	437	
9.8 Direct File Organization	442	
9.9 Indexed Sequential Organization	445	
9.9.1 Searching a Record	445	
9.9.2 Addition/Deletion of a Record	446	
9.9.3 Storage Devices for Indexed Sequential Files	447	
9.9.4 Multilevel Indexed Files	448	
9.10 Choice of File Organization	448	
9.11 Graded Problems	451	
Chapter 10: Advanced Data Structures	459	
10.1 AVL Trees	459	
10.1.1 Searching an AVL Tree	461	
10.1.2 Inserting a Node in an AVL Tree	462	
10.2 Sets	468	
10.2.1 Representation of Sets	469	
10.2.2 Operations on Sets	470	
10.2.3 Applications of Sets	476	
10.3 Skip Lists	478	
10.4 B-Trees	480	
10.4.1 Searching a Key in a B-Tree	482	
10.4.2 Inserting a Key in a B-Tree	483	
10.4.3 Deleting a Key from a B-Tree	484	
10.4.4 Advantages of B-Trees	487	

10.5 Searching by Hashing	489	
10.5.1 Types of Hashing Functions	490	
10.5.2 Requirements for Hashing Algorithms		491
10.5.3 Overflow Management (Collision Handling)		491
<i>Appendix A ASCII Codes (Character Sets)</i>		<i>494</i>
<i>Appendix B Table of Format Specifiers</i>		<i>495</i>
<i>Appendix C Escape Sequences</i>	<i>496</i>	
<i>Appendix D Trace of Huffman Algorithm</i>		<i>497</i>
<i>Index</i>	<i>501</i>	