

1. Record Nr.	UNINA9910735400503321
Autore	Kalicharan Noel
Titolo	Advanced topics in C : core concepts in data structures // Noel Kalicharan
Pubbl/distr/stampa	New York : , : Apress, , 2013
ISBN	1-4302-6401-2
Edizione	[1st ed. 2013.]
Descrizione fisica	1 online resource (xviii, 292 pages) : illustrations
Collana	The expert's voice in C Advanced topics in C
Disciplina	0006.312 004 005.133 006.3/12
Soggetti	C (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	<p>""Advanced Topics in C""; ""Contents at a Glance""; ""Contents""; ""About the Author""; ""About the Technical Reviewer""; ""Preface""; ""Chapter 1: Sorting, Searching, and Merging""; ""1.1 Sorting an Array: Selection Sort""; ""1.1.1 Analysis of Selection Sort""; ""1.2 Sorting an Array: Insertion Sort""; ""1.2.1 Analysis of Insertion Sort""; ""1.3 Inserting an Element in Place""; ""1.4 Sorting an Array of Strings""; ""1.5 Sorting Parallel Arrays""; ""1.6 Binary Search""; ""1.7 Searching an Array of Strings""; ""1.8 Example: Word Frequency Count""; ""1.9 Merging Ordered Lists""</p> <p>""1.9.1 Implementing the Merge""""Chapter 2: Structures""; ""2.1 Defining Structures""; ""2.2 How to Declare a Structure""; ""2.2.1 typedef""; ""2.3 Working with an Array of Structures""; ""2.4 Searching an Array of Structures""; ""2.5 Sorting an Array of Structures""; ""2.6 How to Read, Search, and Sort a Structure""; ""2.7 Nested Structures""; ""2.8 Working with Fractions""; ""2.8.1 Manipulating Fractions""; ""2.9 A Voting Problem""; ""2.10 Passing Structures to Functions""; ""Chapter 3: Pointers""; ""3.1 Defining Pointers""; ""3.2 Passing Pointers as Arguments""</p> <p>""3.3 More on Passing an Array as an Argument""""3.4 Character Pointers""; ""3.5 Pointer Arithmetic""; ""3.6 Pointers to Structures""; ""3.7 Pointers to Functions""; ""3.8 Void Pointers""; ""Chapter 4: Linked</p>

Lists"; "4.1 Defining Linked Lists"; "4.2 Basic Operations on a Linked List"; "4.2.1 Counting the Nodes in a Linked List"; "4.2.2 Searching a Linked List"; "4.2.3 Finding the Last Node in a Linked List"; "4.3 Dynamic Storage Allocation: malloc, calloc, sizeof, free"; "4.3.1 malloc"; "4.3.2 calloc"; "4.3.3 sizeof"; "4.3.4 free"  
"4.4 Building a Linked List: Adding New Item at the Tail""4.5 Insertion into a Linked List"; "4.6 Building a Linked List: Adding a New Item at the Head"; "4.7 Deletion from a Linked List"; "4.8 Building a Sorted Linked List"; "4.9 Example: Palindrome"; "4.10 Saving a Linked List"; "4.11 Arrays vs. Linked Lists"; "4.12 Storing a Linked List Using Arrays"; "4.13 Merging Two Sorted Linked Lists"; "4.14 Circular and Two-Way Linked Lists"; "4.14.1 Circular Lists"; "4.14.2 Two-Way (Doubly Linked) Lists"; "Chapter 5: Stacks and Queues"; "5.1 Abstract Data Types"  
"5.2 Stacks""5.2.1 Implementing a Stack Using an Array"; "5.2.2 Implementing a Stack Using a Linked List"; "5.3 Creating a Stack Header File"; "5.4 A General Stack Type"; "5.4.1 Example: Convert from Decimal to Binary"; "5.5 Converting Infix to Postfix"; "5.5.1 Evaluating a Postfix Expression"; "5.6 Queues"; "5.6.1 Implementing a Queue Using an Array"; "5.6.2 Implementing a Queue Using a Linked List"; "Chapter 6: Recursion"; "6.1 Recursive Definition"; "6.2 Writing Recursive Functions in C"; "6.3 Converting a Decimal Number to Binary Using Recursion"  
"6.4 Printing a Linked List in Reverse Order"

---

## Sommario/riassunto

C is the most widely used programming language of all time. It has been used to create almost every category of software imaginable and the list keeps growing every day. Cutting-edge applications, such as Arduino, embeddable and wearable computing are ready-made for C. Advanced Topics In C teaches concepts that any budding programmer should know. You'll delve into topics such as sorting, searching, merging, recursion, random numbers and simulation, among others. You will increase the range of problems you can solve when you learn how to manipulate versatile and popular data structures such as binary trees and hash tables. This book assumes you have a working knowledge of basic programming concepts such as variables, constants, assignment, selection (if..else) and looping (while, for). It also assumes you are comfortable with writing functions and working with arrays. If you study this book carefully and do the exercises conscientiously, you would become a better and more agile programmer, more prepared to code today's applications (such as the Internet of Things) in C.

---