

Helmut Herold

C-Kompaktreferenz

Inhalt

I	Einleitung	I
1.1	Inhalt	1
1.2	Hinweis auf detailliertere Lern- und Lehrunterlagen	2
2	C-Theorie	5
2.1	Typische Wertebereiche für die einzelnen Datentypen	5
2.2	Prioritätstabelle für Operatoren	5
2.3	Regeln für C-Variablenamen	6
2.4	Die von C reservierten Schlüsselwörter	6
2.5	Verschiedene Arten von C-Konstanten	6
2.6	Symbolische Konstanten	7
2.7	Einfache Makros	8
2.8	Ausdrücke und Operatoren	8
2.8.1	TRUE und FALSE in C	9
2.8.2	Keine unnötige Auswertung rechts von && und	9
2.8.3	Erlaubte/Unerlaubte Operationen für einfache Datentypen	9
2.9	Zeiger auf Konstanten und konstante Zeiger	9
2.10	Ein- und Ausgabe eines Zeichens	9
2.11	Der sizeof-Operator	10
2.12	Ausgabe mit printf	11
2.12.1	Steuerzeichen	11
2.12.2	Umwandlungsvorgaben	11
2.13	Eingabe mit scanf	14
2.14	Datentypumwandlungen	17
2.14.1	Implizite Datentypumwandlungen	17
2.14.2	Explizite Datentypumwandlungen mit cast-Operator	19
2.15	Anweisungen und Blöcke	20
2.16	Die if-Anweisung	20
2.16.1	Einseitige if-Anweisung	20
2.16.2	Zweiseitige if-Anweisung	21
2.17	Die bedingte Bewertung	22
2.18	Die switch-Anweisung	22
2.19	Der Komma-Operator	24
2.20	Die for-Anweisung	24
2.21	Die while-Anweisung	25

2.22	Die do...while-Anweisung	27
2.23	Die break-Anweisung	28
2.24	Die continue-Anweisung	28
2.25	Marken und goto-Anweisung	28
2.26	Die Konstante EOF	29
2.27	Zufallszahlen in C	29
2.28	Funktionen	31
2.28.1	Definition von Funktionen	31
2.28.2	Die return-Anweisung	33
2.28.3	Funktionen ohne Rückgabewert	34
2.28.4	FORWARD-Deklarationen	35
2.28.5	Funktionsprototypen	36
2.28.6	Implizite Datentypumwandlung beim Funktionsaufruf	37
2.28.7	Leere Parameterliste durch die Angabe von void	37
2.28.8	Call by value	37
2.28.9	Call by reference	38
2.28.10	Argumentauswertung findet vor Funktionsaufruf statt	39
2.28.11	Reihenfolge der Argumentauswertung ist nicht festgelegt	39
2.28.12	Aufrufreihenfolge von Funktionen in Ausdrücken	41
2.28.13	Ellipsen-Prototypen	42
2.29	Verfahren zum Abarbeiten variabel langer Argumentlisten	42
2.29.1	Argumentzahl wird beim Aufruf als erstes Argument übergeben	42
2.29.2	Ende der Argumentliste wird durch speziellen Wert gekennzeichnet	43
2.29.3	Kennzeichen jedes Element-Typs durch vorangestelltes Formatelement	44
2.29.4	Aufruf der Funktion vprintf	44
2.30	Rekursive Funktionen	45
2.31	Zeiger auf Funktionen	47
2.32	Gültigkeitsbereich, Lebensdauer und Speicherort	48
2.32.1	Gültigkeitsbereich	48
2.32.2	Lebensdauer	51
2.32.3	Speicherort	52
2.32.4	Das Schlüsselwort extern	52
2.32.5	Das Schlüsselwort auto (überflüssig)	52
2.32.6	Das Schlüsselwort static	52
2.32.7	Das Schlüsselwort register	52
2.33	Die Schlüsselwörter const und volatile	53
2.33.1	Das Schlüsselwort const	53
2.33.2	Das Schlüsselwort volatile	54
2.33.3	Kombination von const und volatile	54
2.34	Arrays und Zeiger	54
2.34.1	Eindimensionale Arrays	54
2.34.2	Mehrdimensionale Arrays	56
2.34.3	Arrayinitialisierungen	58
2.34.4	Arrays als Parameter von Funktionen	63
2.34.5	sizeof-Operator liefert die Größe eines Arrays	65
2.34.6	Gegenüberstellung von Arrays und Zeigern	66
2.34.7	Erlaubte Operationen mit Zeigern	69
2.34.8	Unerlaubte Operationen mit Zeigern	69
2.34.9	Zeigerarrays	70

2.34.10	Zeiger auf Arrays	71
2.34.11	Zeigerzeiger	72
2.34.12	Zeigerarrays mit Funktionsadressen	72
2.35	Strings und char-Zeiger	74
2.35.1	Unterschied zwischen char-Zeiger und char-Array	75
2.35.2	Funktionen zum Umgang mit Strings (<string.h>)	75
2.35.3	Umwandeln von Strings in numerische Werte	77
2.35.4	Umwandeln von numerischen Werten in Strings	78
2.35.5	Besonderheiten beim Einlesen von Strings	78
2.36	Erstellen und Verstehen komplexer Datentypen	79
2.36.1	Bottom-up-Entflechten komplexer Datentypen	79
2.36.2	Top-Down-Flechten komplexer Datentypen	80
2.37	Präprozessordirektiven	81
2.37.1	Präprozessordirektiven zur bedingten Kompilierung	81
2.37.2	Einkopieren von anderen Dateien (#include)	82
2.37.3	Konstanten- und Makrodefinitionen (#define und #undef)	83
2.37.4	Unterschiede zwischen Funktionen und Makros	87
2.37.5	Vordefinierte Makronamen	91
2.37.6	Die restlichen Präprozessordirektiven	92
2.38	Argumente auf der Kommandozeile	93
2.38.1	Einige Besonderheiten	94
2.38.2	Abarbeiten von Optionen in der Kommandozeile	95
2.39	Dynamische Speicherreservierung	96
2.39.1	Dynamische Speicherreservierung mit malloc	96
2.39.2	Dynamische Arrays für beliebige Datentypen	97
2.39.3	Konvertierung von void-Zeigern	99
2.39.4	Speicher reservieren und initialisieren mit calloc	99
2.39.5	Freigeben von dynamisch reserviertem Speicher mit free	100
2.39.6	Verändern der Größe dynamischer Speicherbereiche mit realloc	101
2.39.7	Schnellere Programme mit größeren Speicherblöcken	102
2.39.8	Dynamische Zeigerarrays	103
2.40	Strukturen	105
2.40.1	Deklaration von Strukturen	105
2.40.2	Definition von Strukturvariablen	106
2.40.3	Zusammenfassung von Strukturdeklaration und -definition	106
2.40.4	Zugriff auf Strukturkomponenten mittels Punktoperator	108
2.40.5	Zuweisung zwischen Strukturkomponenten	109
2.40.6	Zuweisung kompletter Strukturvariablen	109
2.40.7	Vergleich von Strukturvariablen ist nicht möglich	110
2.40.8	Casting für komplette Strukturvariable ist nicht möglich	110
2.40.9	Adreß- und sizeof-Operator sind für Strukturvariablen erlaubt	110
2.40.10	Initialisierung von Strukturvariablen	112
2.40.11	Strukturarrays	113
2.40.12	Strukturen als Funktionsparameter	114
2.40.13	Zugriff auf Strukturkomponenten mittels Rückgabewert	115
2.40.14	Zeiger auf Strukturen	115
2.40.15	Unterschied zwischen Strukturzeiger und Zeiger in Struktur	115
2.40.16	Pfeiloperator für einen Zugriff über Strukturzeiger	116
2.40.17	Kombination von Pfeiloperator mit Postfix/Präfix-Operatoren	117

2.40.18	Strukturzeiger reservieren keinen Speicherplatz	118
2.40.19	Dynamische Arrays von Strukturzeigern	118
2.41	Unions	122
2.41.1	Speicherplatzeinsparung durch Unions	124
2.41.2	Overlay-Technik zur unterschiedlichen Interpretation von Daten	124
2.42	Bitfelder	125
2.42.1	Namenlose Bitfelder	127
2.42.2	Regeln für Bitfelder	127
2.43	Definition eigener Datentypnamen mit typedef	129
2.43.1	Mehrere Namen mit einer typedef-Definition	130
2.43.2	Höhere Portabilität und bessere Lesbarkeit durch typedef	131
2.44	Definition eigener Datentypen mit enum	132
2.44.1	Regeln für enum	134
2.45	Dateien	136
2.45.1	Höhere E/A-Funktionen (<stdio.h>)	136
2.45.2	Elementare E/A-Funktionen	165
2.46	Dateien, Directories und ihre Attribute	176
2.46.1	Struktur stat – Attribute zu einer Datei	176
2.46.2	stat und fstat – Erfragen von Dateiattributen	176
2.46.3	Dateiarten	177
2.46.4	Zugriffsrechte einer Datei	177
2.46.5	chmod – Ändern der Zugriffsrechte für eine Datei	178
2.46.6	access – Prüfen der Zugriffsrechte für eine Datei	180
2.46.7	Größe einer Datei	180
2.46.8	Zeiten einer Datei	181
2.46.9	mkdir – Anlegen eines neuen Directories	181
2.46.10	rmdir – Löschen eines leeren Directories	182
2.46.11	chdir – Wechseln in ein neues Directory	182
2.46.12	getcwd – Erfragen des Working-Directory-Pfadnamens	182
2.46.13	struct dirent – Aufbau eines Eintrags in einer Directory-Datei	182
2.46.14	opendir, readdir, rewinddir und closedir – Lesen von Directories	183
2.46.15	Durchlaufen eines ganzen Directorybaums	183
2.46.16	Gerätedateien	186
2.46.17	Anormale Beendigung mit Funktion abort	188
2.46.18	Anhalten eines Prozesses mit der Funktion sleep	188
2.47	Umgebung eines ablaufenden Programms	189
2.47.1	Startup-Routine – Startadresse eines Programms	189
2.47.2	main – Benutzerdefinierter Startpunkt eines Programms	189
2.47.3	Beendigung eines Prozesses	189
2.47.4	Exit-Status eines Prozesses	190
2.47.5	Normales Beenden der Funktion main mit return	190
2.47.6	exit – Normales Beenden eines Programms mit cleanup	190
2.47.7	_exit – Normales Beenden eines Programms ohne cleanup	191
2.47.8	atexit – Einrichten von Exithandlern	191
2.47.9	Start und Beendigung eines Prozesses im Überblick	191
2.48	Environment eines Prozesses	192
2.48.1	Environment-Liste	193
2.48.2	Zugriff auf die ganze Environment-Liste	193
2.48.3	getenv – Erfragen einzelner Environment-Variablen	194

2.48.4	putenv, setenv und unsetenv (Linux/Unix) – Ändern, Hinzufügen oder Löschen von Environment-Variablen	195
2.49	Starten eines anderen Programms	195
2.49.1	Die Funktion system	195
2.49.2	Die exec-Funktionen	196
2.50	Signale	199
2.50.1	Das Signalkonzept und die Funktion signal	199
2.50.2	signal – Einrichten von Signalhandlern	199
2.50.3	Signale und die exec-Funktionen	200
2.50.4	Signalnamen und Signalnummern	201
2.51	Nichtlokale Sprünge	202
2.51.1	setjmp und longjmp – Springen über Funktionsgrenzen	203
2.51.2	Der Datentyp jmp_buf	204
2.51.3	setjmp – Einrichten eines Anspringpunktes	204
2.51.4	longjmp – Sprung zu einem mit setjmp markierten Punkt	205
2.51.5	Rückkehr über mehrere Ebenen bei bestimmten Ereignissen	205
2.51.6	auto-, register-, static- und volatile-Variable bei longjmp	206
3	Headerdateien	207
3.1	Überblick über die Headerdateien	207
3.2	Primitive Systemdatentypen	207
3.3	Limits, Konstanten	208
3.4	Von ANSI C vordefinierte Makros	209
3.5	<stdarg.h> – Abarbeiten variabler Argumentlisten	209
3.6	<ctype.h> – Klassifizieren oder Umwandeln von Zeichen	210
3.7	<assert.h> – Testmöglichkeit mit der assert-Funktion	211
3.8	<errno.h> – Anzeigen von Fehlern in Bibliotheksfunktionen	211
3.9	<limits.h> – Limits für ganzzahlige Datentypen	212
3.10	<float.h> – Limits/Eigenschaften für Gleitpunktdatentypen	212
3.11	<math.h> – Mathematische Funktionen	213
3.12	<stddef.h> – Standarddefinitionen	215
3.13	<stdlib.h> – Allgemein nützliche Funktionen	215
3.13.1	Allokieren und Freigeben von Speicherplatz	215
3.13.2	Environment-Variablen	216
3.13.3	Programmbeendigung	216
3.13.4	Zufallszahlen	216
3.13.5	Absolutwerte	217
3.13.6	Binäre Suche und Quicksort	217
3.13.7	Quotient und Rest einer Division	217
3.13.8	Konvertierung von Strings in numerische Werte	217
3.13.9	Vielbyte-Zeichen	218
3.14	<string.h> – Umgang mit Zeichenketten	219
3.15	<stdio.h> – Standard-E/A-Funktionen	223
3.15.1	Öffnen einer Datei	223
3.15.2	Öffnen einer Datei mit bereits existierendem Stream	224
3.15.3	Schließen einer Datei	224
3.15.4	EOF- und Fehler-Flags	224
3.15.5	Löschen des Fehler- und EOF-Flags	225
3.15.6	Lesen eines Zeichens von stdin	225

3.15.7	Schreiben eines Zeichens auf stdout	225
3.15.8	Lesen eines Zeichens aus einer Datei	225
3.15.9	Schreiben eines Zeichens in eine Datei	225
3.15.10	Zurückschieben eines gelesenen Zeichens in den Eingabepuffer	225
3.15.11	Lesen einer ganzen Zeile von stdin	226
3.15.12	Lesen einer ganzen Zeile aus einer Datei	226
3.15.13	Schreiben einer ganzen Zeile auf stdout	226
3.15.14	Schreiben einer ganzen Zeile in eine Datei	226
3.15.15	Binäres Lesen und Schreiben ganzer Blöcke	226
3.15.16	Formatiertes Lesen von stdin	227
3.15.17	Formatiertes Lesen aus einer Datei	230
3.15.18	Formatiertes Lesen aus einem String	230
3.15.19	Pufferung	230
3.15.20	Formatiertes Schreiben auf stdout	232
3.15.21	Formatiertes Schreiben in Datei	236
3.15.22	Formatiertes Schreiben in einen String	236
3.15.23	Formatiertes Schreiben auf stdout (Argumentzeiger)	237
3.15.24	Formatiertes Schreiben in einen String (Argumentzeiger)	237
3.15.25	Positionieren in Dateien	237
3.15.26	Löschen und Umbenennen von Dateien	239
3.15.27	Temporäre Dateien	239
3.15.28	Ausgabe von Systemfehlermeldungen	240
3.15.29	Filedeskriptoren und der Datentyp FILE	241
3.16	<dirent.h> – Lesen von Directories	242
3.16.1	Aufbau eines Eintrags in einer Directorydatei (struct dirent)	242
3.16.2	Lesen von Directories	242
3.17	<time.h> – Datums- und Zeitangaben	243
3.17.1	Datentypen	243
3.17.2	Konstanten	244
3.17.3	Datums- und Zeitfunktionen	244
3.17.4	Zusammenfassung der wichtigsten Zeitformatumwandlungen	247
3.18	<signal.h> – Einrichten von Signalhandlern	248
3.18.1	Das Signalkonzept und die Funktion signal	248
3.18.2	signal – Einrichten von Signalhandlern	248
3.18.3	Signale und die exec-Funktionen	249
3.18.4	Signalnamen und Signalnummern	249
3.19	<setjmp.h> – Nichtlokale Sprünge	251
3.19.1	setjmp und longjmp – Springen über Funktionsgrenzen	252
3.19.2	Der Datentyp jmp_buf	253
3.19.3	setjmp – Einrichten eines Ansprungspunktes	253
3.19.4	longjmp – Sprung zu einem mit setjmp markierten Punkt	254
3.19.5	Rückkehr über mehrere Ebenen bei bestimmten Ereignissen	254
3.19.6	auto-, register-, static- und volatile-Variable bei longjmp	255
4	Wichtige Algorithmen	257
4.1	Sortieralgorithmen	257
4.1.1	Bubble-Sort	257
4.1.2	Insert-Sort	258

4.1.3	Select-Sort	260
4.1.4	Merge-Sort	262
4.1.5	Quick-Sort	263
4.1.6	Verwendung der Bibliotheksfunktion qsort	265
4.2	Suchalgorithmen	273
4.2.1	Binäre Suche	273
4.2.2	Verwendung der Bibliotheksfunktion bsearch	275
4.2.3	Suchen in Texten	276
4.3	Stapel	281
4.4	Schlangen	282
4.5	Kombinatorik (Permutationen)	283
4.5.1	Permutationen	283
4.5.2	Zufallspermutationen	289
4.5.3	k-Teilmengen	290
4.5.4	Potenzmenge	292
4.5.5	k-Permutationen	293
4.5.6	k-Tupel	294
4.5.7	Binär-Tupel	295
4.5.8	"k aus n"-Code	296
4.5.9	Gray-Code	298
4.5.10	Partitionen	299
4.6	Einige Anwendungen für rekursive Funktionen	301
4.6.1	Rekursion bei Baumstrukturen	301
4.6.2	Rekursion bei rekursiv definierten mathematischen Funktionen	302
4.6.3	Rekursives Umwandeln einer Dezimalzahl in eine Dualzahl	305
4.6.4	Rekursives Füllen von beliebigen Figuren (Grafik)	306
4.6.5	Rekursion bei Fraktalen (Lindenmayer-Systeme)	307
4.6.6	Rekursion bei der Syntaxanalyse	309
4.6.7	Rekursion beim Backtracking-Verfahren	313
4.7	Rekursive Strukturen	315
4.7.1	Einfach verkettete Listen	315
4.7.2	Sortierte Listen und Operationen für einfach verkettete Listen	319
4.7.3	Doppelt verkettete Listen	322
4.7.4	Ringliste (Beispiel: Das Josephus-Spiel)	325
4.7.5	Mischen von zwei sortierten Listen	327
4.7.6	Eine einzige Liste, aber unterschiedlich sortiert	331
4.7.7	Binärbäume	334
4.8	Kalenderalgorithmen	341
4.8.1	Wochentag eines Datums	341
4.8.2	Julianischer Tag	343
4.8.3	Gregorianisches Datum	343
4.8.4	Nummer eines Tages im Jahr	345
4.8.5	Ostersonntag	345
4.9	Schnelles Potenzieren mit Legendre	348
4.10	Quadratwurzel mit Newton-Iteration berechnen	348
4.11	Aufzählbare Mengen	351
4.12	Lateinische Quadrate	351
4.13	Magische Quadrate	352
4.13.1	Magische Quadrate für ungerades n ($n \bmod 2 = 1$)	353

4.13.2	Magische Quadrate für durch 4 teilbare n ($n \bmod 4 = 0$)	353
4.13.3	Magische Quadrate für $n \bmod 4 = 2$	353
4.14	Primzahlen	354
4.14.1	Primzahlen mit dem Sieb des Eratosthenes	354
4.14.2	Primzahltest von Solovay / Strassen (Stochastischer Algorithmus)	355
4.15	Gaußsche Eliminationsverfahren	357
4.16	Umwandeln arabischer Zahlen in römische Zahlen	360

5 Nützliche Programme und Hilfsroutinen 363

5.1	Programmierung	363
5.1.1	Feststellen, ob ein String eine erlaubte ganze Zahl enthält	363
5.1.2	Feststellen, ob ein String nur Kleinbuchstaben enthält	363
5.1.3	Feststellen, ob ein String nur Vokale enthält	363
5.1.4	Nützliche mathematische Hilfsroutinen	363
5.1.5	Streichen aller Vorkommen eines Zeichens aus einem String	364
5.1.6	Ersetzen aller Vorkommen eines Wortes in einem String	364
5.1.7	Suchen von Strings in Texten	366
5.1.8	Sichere Zahleneingabe	368
5.1.9	Automatisches "(Ent-)Flechten" von Datentypen	371
5.2	Informatik	375
5.2.1	Duale Darstellung und duale BCD-Darstellung einer Zahl	375
5.2.2	Konvertieren von Dezimalzahlen in Dual-, Oktal- und Hexadezimalsystem	377
5.2.3	Konvertierung eines Dezimalbruchs in ein beliebiges System	379
5.2.4	Die Türme von Hanoi	380
5.2.5	Hammingabstand eines Codes und das Korrigieren von Fehlern	382
5.2.6	Fano-Bedingung (für Decodierbarkeit eines Codes)	387
5.3	Mathematik	390
5.3.1	Division mit beliebiger Genauigkeit	390
5.4	Wirtschaft	392
5.4.1	Addieren von deutschen Kommazahlen	392
5.4.2	Verlauf eines Sparkapitals berechnen und ausgeben	393
5.4.3	Erforderliche Sparrate für ein gewünschtes Endkapital berechnen	395
5.4.4	Hypothek	396
5.4.5	Darlehen	399
5.5	Allgemein	402
5.5.1	Umrechnungen für einen Urlaub in den USA	402
5.5.2	Ein Dauerkalender	406
5.5.3	Iterierte Funktionssysteme	409

6 Grafik 413

6.1	Semigrafik in Turbo- und Borland-C (Headerdatei conio.h)	413
6.1.1	Bildschirmkoordinaten	413
6.1.2	Farben	413
6.1.3	Semigrafikfunktionen	413
6.1.4	Wichtige Tastencodes (für Borland-/Turbo-C)	417
6.2	Semigrafik unter Linux/Unix (Headerdatei curses.h)	419
6.2.1	Allgemeines	419
6.2.2	Bildschirmkoordinaten	420

6.2.3	Semigrafikfunktionen	420
6.2.4	Funktions- und Positionierungstasten	423
6.3	Vollgrafik	424
6.3.1	Bildschirmkoordinaten	424
6.3.2	Farben	424
6.3.3	Vollgrafikfunktionen	425
6.4	Transformation mathematischer Koordinaten in Bildschirmkoordinaten	438
6.4.1	Maßstabsfaktoren	438
6.4.2	Punkttransformation	438
6.4.3	Nullpunkt	439
6.5	Programmierung der Maus unter DOS	439
6.5.1	Funktionen	440
6.5.2	Verschiedene Formen des Mausursors	440
A	Die ASCII-Tabelle	447
B	Bestelldaten für die vollständigen Lehr- und Lernunterlagen	451
	Index	453