LERNEN EINFACH GEMACHT

3. Auflage

C

für dummies®

C-Programmierung schnell verstehen und umsetzen

Clever editieren, kompilieren und debuggen

Den sicheren Umgang mit Variablen, Funktionen, Schleifen und Zeigern lernen

Dan Gookin

C für Dummies

Schummelseite

NUMERISCHE DATENTYPEN

Die folgende Tabelle zeigt C-Schlüsselwörter für numerische Daten, ihre Variablentypen und ihre Bereiche:

Schlüsselwort	*Variablentyp*	*Bereich*	*Benötigter Speicherplatz*
`char`	Zeichen	-128 bis 127 oder 0 bis 255, je nach Compiler	1 Byte

`short`
oder
`short int`
	kurze Ganzzahl	-32.768 bis 32.767	2 Bytes

`int`	Ganzzahl	-2.147.483.648 bis 2.147.483.647	4 Bytes

`long`
oder
`long int`
	lange Ganzzahl	-2.147.483.648 bis 2.147.483.647	4 Bytes

`long long`
oder
`long long int`
	sehr lange Ganzzahl	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807	8 Bytes

`unsigned char`	vorzeichenloses Byte	0 bis 255	1 Byte
`unsigned short`	vorzeichenlose kurze Ganzzahl	0 bis 65.535	2 Bytes
`unsigned int`	vorzeichenlose Ganzzahl	0 bis 4.294.967.295	4 Bytes
`unsigned long`	vorzeichenlose lange Ganzzahl	0 bis 4.294.967.295	4 Bytes

`unsigned long long`
oder
`unsigned long long int`
	vorzeichenlose sehr lange Ganzzahl	0 bis 18.446.744.073.709.551.615	8 Bytes

Schlüsselwort	*Variablentyp*	*Bereich*	*Benötigter Speicherplatz*
`float`	Fließkommazahl niedriger Genauigkeit (circa 7 Stellen)	$\pm 3{,}4 \times 10^{-38} \pm 3{,}4 \times 10^{38}$	4 Bytes
`double`	Fließkommazahl doppelter Genauigkeit (circa 15 Stellen)	$\pm 1{,}7 \times 10^{-308} \pm 1{,}7 \times 10^{308}$	8 Bytes

ESCAPE-SEQUENZEN

Die Programmierung in C ist schnell – Sie müssen nur wenig tippen – oft nur zweimal –, um einen Tabulator, eine neue Zeile, ein Fragezeichen und mehr zu erhalten. Die folgende Tabelle zeigt die Sequenzen, die Sie für eine Vielzahl von Aufgaben benötigen:

Sequenz	*Bedeutung*
`\a`	Der Lautsprecher gibt einen Ton aus.
`\b`	Backspace (der Cursor wird ein Zeichen nach links gebracht)
`\f`	Blatt auswerfen (beim Drucker), auf dem Bildschirm wird je nach Konfiguration/Betriebssystem der Bildschirm gelöscht oder einfach nur ein seltsames Zeichen ausgegeben.
`\n`	neue Zeile, wirkt wie Drücken der ↵-Taste
`\r`	Wagenrücklauf, der Cursor wird wieder an den Zeilenanfang gesetzt.
`\t`	Tabulator ansteuern
`\v`	vertikaler Tabulator (setzt den Cursor in eine neue Zeile, je nach Konfiguration/Betriebssystem wird auch einfach nur ein seltsames Zeichen ausgegeben)
`\\`	das Backslash-Zeichen `\`
`\'`	der Apostroph `'`
`\"`	das Anführungszeichen `"`
`\?`	das Fragezeichen `?`
`\0`	das »Null«-Byte (das ist eine Null hier)

Sequenz	*Bedeutung*
`\nnn`	ein Zeichenwert als Oktalwert (zur Basis 8)
`\xnnn`	ein Zeichenwert als Hexadezimalzahl (zur Basis 16)
`\Xnnn`	Wie kleines `x` in `\xnnn`, aber die Hex-Ziffern werden großgeschrieben.
`\x084`	kleines »ä« (nur unter Windows-Kommandozeile)
`\x094`	kleines »ö« (nur unter Windows-Kommandozeile)
`\x081`	kleines »ü« (nur unter Windows-Kommandozeile)
`\x08e`	großes »Ä« (nur unter Windows-Kommandozeile)
`\x099`	großes »Ö« (nur unter Windows-Kommandozeile)
`\x09a`	großes »Ü« (nur unter Windows-Kommandozeile)
`\x0e1`	scharfes »ß« (nur unter Windows-Kommandozeile)

KONVERTIERUNGSZEICHEN VON `PRINTF`

Wenn Sie in C programmieren, verwenden Sie Konvertierungszeichen – meist das Prozentzeichen und einen Buchstaben – als Platzhalter für Variablen, die Sie darstellen möchten. Die folgende Tabelle zeigt die Konvertierungszeichen und ihre Darstellung:

Konvertierungszeichen	*Argument (Variableninhalt) darstellen als*
`%c`	einzelnes Zeichen
`%d`	Dezimalzahl (`int`) mit Vorzeichen
`%e`	Fließkommazahl mit Vorzeichen in Exponentialschreibweise
`%f`	Fließkommazahl mit Vorzeichen (`float`)
`%g`	Fließkommazahl, die entweder als `%e` oder `%f` dargestellt wird, je nachdem, was kürzer ist
`%i`	Dezimalzahl (`int`) mit Vorzeichen (wie `%d`)
`%o`	Oktalzahl (zur Basis 8) ohne Vorzeichen (`int`)
`%s`	String
`%u`	Dezimalzahl (`int`) ohne Vorzeichen

Konvertierungszeichen	*Argument (Variableninhalt) darstellen als*
`%x`	Hexadezimalzahl (zur Basis 16) ohne Vorzeichen (`int`)
%p	Dient zur Ausgabe von Zeigern, also Adressen im Speicher. Siehe Kapitel 12.

ERLAUBTE SYMBOLE BEI VERGLEICHEN

Wenn Sie Programme in C schreiben, müssen Sie Vergleichssymbole verwenden. Die in C verwendeten Symbole, ihre Bedeutungen und Beispiele sind in der folgenden Tabelle aufgeführt:

Vergleich	*Bedeutung*	*»wahre« Beispiele*
`<`	kleiner als	`1 < 58 < 9`
`==`	gleich	`5 == 50 == 0`
`>`	größer als	`8 > 510 > 0`
`<=`	kleiner oder gleich	`4 <= 58 <= 8`
`>=`	größer oder gleich	`9 >= -52 >= 2`
`!=`	ungleich	`1 != 04 != 3.99`

ZEIGER, KLAMMERN UND ARITHMETIK

Situation	*Speicheradresse*	*Wert*
`p`	selbst	-
`*p`	-	selbst
`*p++`	inkrementiert nach dem Lesen des Wertes	unverändert
`*(p++)`	inkrementiert nach dem Lesen des Wertes	unverändert
`(*p)++`	unverändert	inkrementiert, nach der Benutzung
`*++p`	inkrementiert vor dem Lesen des Wertes	unverändert

Situation	*Speicheradresse*	*Wert*
`*(++p)`	inkrementiert vor dem Lesen des Wertes	unverändert
`++*p`	unverändert	inkrementiert vor der Benutzung
`++(*p)`	unverändert	inkrementiert vor der Benutzung
`p*++`	kein Zeiger	kein Zeiger
`p++*`	kein Zeiger	kein Zeiger

ZEIGER – TYPISCHE ANWENDUNGEN

Zuerst Zeiger des richtigen Typs erzeugen:

```
float* f;
```

Dann die Adresse einer Variablen zuweisen:

```
f = &andere_variable;
```

Nun den Zeiger benutzen:

```
printf("%0.f\n", *f);
```

Ohne Sternchen `*` stellt der Zeiger eine Speicheradresse dar.

Mit Sternchen `*` bezeichnet der Zeiger den Inhalt der Speicheradresse.

Zeiger - und Variablentyp müssen immer zusammenpassen: `float` gehört zu `float*`, `int` zu `int*` und so weiter.

Wichtig: Zeiger müssen *vor* der Benutzung immer zuerst mit einem Wert (also einer Speicheradresse) initialisiert werden. Dies kann durch den Adressoperator `&` oder durch die Funktion `malloc` geschehen.

ZEIGER UND ARRAY-KLAMMERN

Array-Notation	*Zeiger-Notation*
`array[0]`	`*a`
`array[1]`	`*(a+1)`

Array-Notation	*Zeiger-Notation*
`array[2]`	`*(a+2)`
`array[3]`	`*(a+3)`
`array[x]`	`*(a+x)`

ZUGRIFFSMODI FÜR `FOPEN`

Modus	*Öffnet eine Datei zum*	*Erzeugt eine Datei?*	*Existierende Datei wird*
`"r"`	Lesen	Nein	geöffnet; Fehler, wenn sie nicht gefunden wird
`"w"`	Schreiben	Ja	überschrieben
`"a"`	Schreiben am Dateiende	Ja	vergrößert; es wird angehängt
`"r+"`	Lesen und Schreiben	Nein	geöffnet; Fehler, wenn sie nicht gefunden wird
`"w+"`	Lesen und Schreiben	Ja	überschrieben
`"a+"`	Lesen und Anhängen	Ja	vergrößert; es wird angehängt

SCHLÜPFRIGE SCHLEIFEN UND IHR ZUSAMMENHANG

for

```
for (startanweisung; bedingung; endanweisung)

 {
    tue(etwas);
 }
```

while

```
 while (bedingung)
 { // der Rumpf wird vielleicht nie ausgeführt
```

```
   tue(etwas);
}
```

do-while

```
do
{ // der Rumpf wird mindestens einmal ausgeführt
   tue(etwas);
} while (bedingung);
```

Dan Gookin

C für dummies®

3. Auflage

Übersetzung aus dem Amerikanischen
von Marcus Bäckmann

WILEY
WILEY-VCH GmbH

C für Dummies

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

3. Auflage 2021

Richtigkeit von Angaben, Hinweisen und Ratschlägen sowie eventuelle Druckfehler keine Haftung.

Coverfoto: © Surf Ink – stock.adobe.com
Korrektur: Petra Heubach-Erdmann, Düsseldorf

Print ISBN: 978-3-527-71845-0
ePub ISBN: 978-3-527-83338-2

Über den Autor

Dan Gookin schreibt schon seit fast drei Jahrzehnten über Technik. Bei ihm verbindet sich seine Liebe zur Schriftstellerei mit der Faszination, Bücher zu erstellen, die gleichzeitig informativ und unterhaltsam und keinesfalls langweilig sind. Man kann Dan Gookin zugestehen, dass seine Methode, Computerbücher zu schreiben, funktioniert, nachdem seine über 160 Titel mit einer Auflage von zwölf Millionen inzwischen in über 30 Sprachen übersetzt wurden.

Sein vielleicht bekanntestes Buch ist *DOS for Dummies*, veröffentlicht 1991. Es wurde das sich am schnellsten verkaufende Computerbuch der Welt, einmal übertraf es sogar bei der Anzahl verkaufter Bände pro Woche die Nummer eins der New-York-Times-Bestseller-Liste (leider konnte es als Nachschlagewerk nicht selbst auf diese Liste kommen). Dieses Buch legte den Grundstein für die gesamte *für Dummies*-Serie, die bis heute eine Ausnahmeerscheinung auf dem Fachbüchermarkt ist.

Dan Gookins populärste Titel umfassen *PCs für Dummies*, *Laptops für Dummies* und *Microsoft Word für Dummies*. Er unterhält zudem die große und hilfreiche (englischsprachige) Website www.wambooli.com.

Dan Gookin hat einen Abschluss in Kommunikation/Bildender Kunst der Universität Kalifornien, San Diego. Er lebt im Nordwesten der USA am Pazifik, wo er gerne Leute nervt – nur wenn sie es verdienen, natürlich.

Inhaltsverzeichnis

Tabellenverzeichnis

Kapitel 10

Kapitel 11

Kapitel 12

Kapitel 13

Kapitel 15

Illustrationsverzeichnis

Kapitel 1

Kapitel 2

Kapitel 3

Kapitel 4

Kapitel 5

Kapitel 6

Kapitel 7

Kapitel 10

Kapitel 11

Kapitel 12

Kapitel 13

Kapitel 16

Kapitel 17

Einleitung

Willkommen bei *C für Dummies* – Ihr erster, x-ter, letzter, aber auf jeden Fall Ihr außerordentlich aussichtsreicher Versuch, die Programmiersprache C zu erlernen.

Obwohl ich nicht garantieren kann, dass Sie ein C-Guru werden, nur weil Sie sich durch dieses Buch kämpfen, so kann ich doch Folgendes mit Sicherheit sagen:

- ✔ Sie werden lernen, wie man ein C-Programm erkennt, und wenn man Ihnen ein Steuerformular Anlage N, die morgendlichen Börsennachrichten, die Bundesligatabelle oder irgendeinen Text in Brailleschrift vorlegt, werden Sie daraus das C-Programm herauspicken können.
- ✔ Sie werden C-Programme schreiben können, die niemals ein Verleger in seinen Büchern abdrucken würde.
- ✔ Sie werden sich über folgenden Gag amüsieren können:

  ```
  while(regen)
    bleib_im:haus();
  ```

  Aber leider wird er Ihnen bei keiner Party helfen, die Leute zu beeindrucken.
- ✔ Sie werden lernen, wie man in C spricht; das heißt, Sie sehen so seltsame Zeichenfolgen wie `printf`, `putchar` und `fgets` und wissen intuitiv, was das Kauderwelsch bedeuten soll.
- ✔ Sie werden viel Spaß haben.

Letzteres kann ich nicht wirklich garantieren. Egal, dieses Buch wurde geschrieben, ohne dass das Damoklesschwert der Mathematik über Ihnen hängt.

Überlassen wir die Wegstreckenoptimierung denjenigen, die sich über der Avogadro-Zahl und der Fibonacci-Reihe den Kopf zerbrechen und über die Vorteile von verschiedenen Codeformatierungen ihrer C-Programme debattieren. Ernsthafte Arbeit ist was für Streber. Lustig wird's, wenn Sie *C für Dummies* lesen.

Was bringt es, C zu lernen?

Schauen Sie auf Ihren Computerbildschirm. Stellen Sie sich vor, etwas bewegt sich darauf. Irgendwas. Wenn man in der Lage ist, Computer zu programmieren, kann man sich etwas vorstellen, und es passiert auf dem Bildschirm. Vielleicht nicht so schnell, wie Sie es wünschen – aber es geht.

Programmierung ist der ultimative Weg, um den Computer zu verstehen. Sie sind gefordert. Sie sagen dem Monster, was es tun soll. Und es wird Ihnen gehorchen, sogar wenn Sie etwas Dummes sagen. Computer sind schnell und gehorsam, aber nicht gerade clever.

Alles, was Ihr Computer tut, jedes Gerät, das an ihm hängt, alles kann durch die Benutzung einer Programmiersprache und das Betätigen der richtigen Hebel manipuliert werden. C ist nicht die am einfachsten zu erlernende Programmiersprache, aber es ist auch nicht die schwierigste. Sie ist populär und gut unterstützt, sodass sie eine gute Wahl darstellt.

Über dieses Buch

Die meisten *… für Dummies*-Bücher sind *Unterhaltungsbücher*. Das ist ein wichtiger Punkt, den ich Leuten vor Augen führe, wenn sie sagen: »Ich habe dieses und jenes in den Dummies-Büchern gelernt.« Keinesfalls! Die *… für Dummies*-Buchreihe ist dazu da, zu erinnern, zu bedienen, zu unterhalten. Sie ist nicht dazu gedacht, etwas zu lehren oder zu lernen (aber der Lernprozess wird natürlich auch nicht behindert).

Wie kann ich also eine Anleitung *C für Dummies* schreiben?

Einfache Antwort: indem ich schwindele. Die meisten Anleitungen setzen voraus, dass man gar nichts weiß. Diese Methode ist gut und schön und so weiter, aber es gibt da einige Stolpersteine. Anstatt Ihnen gleich von allem einen kleinen Bissen zu geben, zwingt man Sie zuerst, Ihren Spinat zu essen, in winzigen, kleinen Portionen. Aber was wollen wir nur mit Spinat? Warum dürfen wir nicht auch mal einen Bissen vom Steak nehmen? Und wo bleibt das Dessert?

Meine Antwort lautet: Sie lernen immer nur ein wenig, aber nie alles auf einmal. Babys werden mit fünf kompletten Sinnen geboren, und dennoch können sie nur so wenig auf einmal lernen. Dieses Buch wird Ihnen ein neues Konzept vorlegen: »Hier ist das Dingsbums! Tipp’s ab und nicke mit dem Kopf. Ich erkläre die Funktionsweise dann später!« Ich denke, dass dies funktioniert, weil Sie sofort interessante Programme erstellen und nicht das Gefühl haben, unnötig lange auf der Stelle zu treten.

Für alle diejenigen, die lieber etwas strukturierter lernen, ist das Buch in 20 Kapitel eingeteilt. Jedes Kapitel enthält eine andere Lektion, jede passend zum Hauptthema. Aber neue Ideen werden vorsichtig

vorgestellt, alle sorgfältig miteinander verknüpft, eben ganz im … *für Dummies*-Stil.

Und obwohl Sie keine Referenz erhalten (trotz der ganzen Tabellen), werden Sie etwas lernen. Also los!

Programme in diesem Buch

Wenn man sich selbst mit einem Buch das Programmieren beibringt, gehört zu dem ganzen Spaß auch das eigenhändige Abtippen der Programme. Auf diese Weise habe ich gelernt zu programmieren. Ich saß mit einer Ausgabe von Dr. David Liens »BASIC lernen mit dem TRS-80« da und 36 Stunden später war ich fertig. Ich schlief. Danach habe ich es noch mal gelesen, weil ich alles vergessen hatte, aber ich erinnerte mich an den Spaß, den ich beim ersten Lesen hatte.

Also müssen Sie Sachen eintippen, das sieht zum Beispiel so aus:

```
Hier bin ich, tipp mich ein. La, la, la.
```

Meistens geben Sie ganze Programme ein, bestehend aus Zeilen ähnlich wie die obige. Tippen Sie alle ein, drücken Sie ↵ am Ende jeder Zeile. Leider ist der Satzspiegel in diesem Buch nicht so breit, daher sind manchmal Zeilen aufgeteilt. So wie hier:

```
Das ist jetzt ein Beispiel für eine wirklich lange Programmzeile,
die von den hinterhältigen Layoutern zerrissen wurde.
```

Wenn Sie so etwas sehen, dann schreiben Sie das nicht in zwei Zeilen. Schreiben Sie am Ende einfach weiter, und alles wird auf den Bildschirm passen. Falls Sie dies vergessen, kommen Ihre Programme durcheinander –

daher habe ich auch einige Erinnerungen im Buch hinterlassen, wenn so etwas passiert.

In diesem Buch sieht die 0 (»Null«) so aus: 0. Geraten Sie nicht in Panik und tippen Sie kein großes O dafür. Es ist nur eine Null.

Ach so, Sie müssen natürlich nicht wirklich alles abtippen. Sie können die Quelltexte auch von der Homepage des Buches herunterladen:

http://www.wiley-vch.de/ISBN9783527718450

Törichte Annahmen über den Leser

Dieses Buch trifft folgende Annahmen über Sie, Ihren PC, Ihren Compiler und – hüstel – Ihren Geisteszustand:

- Sie sollten Erfahrung mit der Kommandozeile (der Shell) haben und sich auf den Laufwerken Ihres PC auskennen. In diesem Buch ist nicht genug Platz, um Ihnen C und den Umgang mit der Kommandozeile beizubringen. Obwohl jeder C-Aspirant in der Regel durch solche simplen Dinge bereits durch ist. Dieser Annahme vertraue ich einfach.
- Sie sollten mit einem Texteditor umgehen können. Ich meine damit kein Word oder eine Textverarbeitung, sondern einen richtigen Editor, wie Notepad++, GNU Emacs oder den Editor, der in Ihrem »C-Programmierpaket« enthalten ist. Ich kann Ihnen nicht den Editor und C beibringen, also lernen Sie den Editor selbst kennen.
- Sie sollten bereits einen Compiler für die Sprache C verfügbar haben. In Kapitel 1 erfahren Sie, woher Sie einen C-Compiler erhalten und wie Sie ihn installieren,

falls Sie noch nicht glücklicher Besitzer eines solchen Compilers sind.

- Die Sache mit dem »plus, plus«, so wie in C++, ist hier kein Thema. Alle Compiler können C++, weil es der Wettbewerb so verlangt. Bevor Sie C++ lernen, sollten Sie ohnehin die Grundlagen von C intus haben, und genau das wird Ihnen hier ja beigebracht. Ein anderes *... für Dummies*-Buch wird Ihnen dann C++ beibringen ... eines Tages.

Wie dieses Buch aufgebaut ist

Damit Sie sich die Themen aufteilen können, ist das Buch in Teile gegliedert, die jeweils einen großen Themenblock behandeln.

Teil I: Das erste Programm

Bevor Sie die Tiefen der Sprache C ausloten, lassen Sie uns zunächst einmal mit dem ersten Programm anfangen – dazu installieren Sie als Erstes den Code::Blocks-Compiler auf Ihrem Rechner, damit Sie sofort loslegen können. Wie bekommen Sie das erste C-Programm überhaupt in den Computer rein? Die ersten Anweisungen helfen Ihnen dabei, zu verstehen, wie man mit der Eingabe und Ausgabe von Daten arbeitet.

Teil II: Grundlegendes Sprachverständnis

Wie bei einer natürlichen Sprache müssen Sie bei einer Programmiersprache zunächst eine grundlegende Vorstellung von den wesentlichen Sprachelementen bekommen. Sie erfahren in diesem Teil, welche wichtigen Kontrollelemente die Sprache C Ihnen

anbietet. Wie können Sie den Programmablauf beeinflussen, wie reagieren Sie auf unterschiedliche Werte, wie bringt man dem Programm bei, sich zu entscheiden oder einen bestimmten Block mehrfach auszuführen?

Teil III: Einfache Datenstrukturen und Zeiger

Datenstrukturen und Algorithmen sind der nächste Themenblock, nachdem Sie sich mit den Kontrollstrukturen befasst haben. Sie lernen die ersten spezielleren Datenspeicher kennen, die Arrays. In diesem Zusammenhang werden Sie erfahren, wie Zeichenketten in C funktionieren und was es mit den berühmt-berüchtigten Zeigern auf sich hat. Nach einem Kapitel über Strukturen und deren dynamische Speicherallozierung kennen Sie alle wichtigen Verfahren zur elementaren Datenspeicherung in C.

Teil IV: Daten speichern und verwalten

Die letzte Herausforderung für den Anfang ist, wie man Programmen beibringt, nach der letzten Anweisung die Daten dauerhaft zu behalten. In diesem Teil erfahren Sie nicht nur, wie man Daten in Dateien speichert und wieder ausliest, sondern wie das Phänomen wachsender Datenmengen in den Griff zu bekommen ist – bisher konnte keines der Programme flexibel auf wachsende Datenbestände reagieren, mithilfe der verketteten Listen werden Sie das meistern.

Teil V: Der Top-Ten-Teil

In diesem Teil habe ich zusammengetragen, was Sie rund ums Thema C wissen sollten. Hier geht es um das

Vermeiden typischer Fehler, um wichtige Webadressen und ein wenig allgemeines Wissen zu C.

Symbole, die in diesem Buch verwendet werden

In diesem Buch werden Sie immer wieder auf Symbole stoßen, die dazu dienen, Sie auf bestimmte Informationen und Sachverhalte aufmerksam zu machen:


Mit diesem Symbol sind technische Informationen gekennzeichnet. Etwas Extrawissen kann nie schaden.


Diese Informationen sollten Sie sich merken.


Diese Informationen sollten Sie unbedingt lesen und im Hinterkopf behalten – das kann Sie vor echten Problemen bewahren.


Hier verrate ich Tipps und Tricks.


Mit diesem Symbol ist Compilerspezifisches gekennzeichnet – hier sollten Sie etwas aufpassen … egal welcher Compiler, es ist immer kritisch.

Schlussgedanken

C zu lernen, ist ein fortwährender Prozess. Nur ein Narr würde sagen, dass er alles über C-Programmierung weiß. Jeden Tag gibt es etwas Neues zu lernen und neue

Anläufe für das gleiche Problem. Nichts ist perfekt, aber viele Ideen kommen dem nahe.

Klar, Leute, die 20 Jahre lang C-Programmierung belegt hatten und viel zu viel Zeit an einer Universität verbrachten, beherrschen C mit einer gewissen Portion Lässigkeit. Egal. Fragen Sie sich selbst: Läuft mein Programm? Okay. Macht es, was es soll? Besser. Genügt es künstlerischen Ansprüchen? Wen kümmert’s? Ich bin zufrieden, wenn Ihr C-Programm läuft. Denken Sie daran: Je mehr Sie lernen, desto besser werden Sie. Sie werden neue Tricks entdecken und Ihren Stil daran anpassen.

Ich hoffe, Sie genießen die Reise, die Sie nun beginnen werden. Spucken Sie in die Hände, starten Sie den Compiler und bereiten Sie sich auf einige gute Stunden Arbeit vor. Sie werden C programmieren!