

EINFÜHRUNG IN DIE PROGRAMMIERUNG  
MIT DER PROGRAMMIERSPRACHE C  
(**EifP**, WS2025/26)

Prof. Dr. Thomas Gabel  
Roman Ahmad, Emre Özöner, Gina Romanazzi

## Aufgabenblatt 9

### Aufgabe 40: Potenzberechnung ohne `pow()` und Summenberechnung ohne `+`

#### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Funktionen in C
- *Rekursion*

#### Aufgabenstellung

a) Schreiben Sie eine

- iterative (rekursionslose)
- *Zusatzaufgabe: rekursive*

C-Funktion, welche die Potenz  $x^y$  von zwei `int`-Parametern  $x$  und  $y$  berechnet. Der Rückgabewert soll vom Typ `double` sein. Achtung: Sie dürfen die durch die C-Mathematikbibliotheken bereitgestellte Funktion `pow()` in Ihren Lösung nicht verwenden. Vergleichen und erklären Sie beide Lösungswege. Die Werte  $x$  und  $y$  sollen vom Benutzer in Ihrem Hauptprogramm eingelesen werden.

b) Schreiben Sie nun eine

- iterative (rekursionslose)
- *Zusatzaufgabe: rekursive*

C-Funktion, welche die Summe  $x + y$  von zwei `int`-Parametern  $x$  und  $y$  berechnet. Der Rückgabewert soll vom Typ `int` sein. Achtung: Sie dürfen den von C bereitgestellten Additionsoperator `+` in Ihrer Lösung nicht verwenden, sondern ausschließlich den Inkrementoperator `++`. Die Werte  $x$  und  $y$  sollen vom Benutzer in Ihrem Hauptprogramm eingelesen werden.

## Aufgabe 41: Primzahltest

### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- scanf, Schleifen, Primzahlen

### Aufgabenstellung

- a) Schreiben Sie ein C-Programm, das bei einer eingegebenen natürlichen Zahl  $n$  ( $n > 1$ ) überprüft, ob es sich um eine Primzahl handelt. Stellen Sie sicher, dass die vom Benutzer eingegebene Zahl auch wirklich eine natürliche Zahl größer als eins ist, indem Sie die Aufforderung zur Eingabe so lange wiederholen, bis diese Bedingung erfüllt ist.
- b) Erweitern Sie Ihr Programm dergestalt, dass die nächstkleinere und nächstgrößere Primzahl zur eingegebenen Zahl ausgegeben werden.

## Aufgabe 42: Aufteilen

### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Felder und Felderindizes
- Schleifen

### Aufgabenstellung

Schreiben Sie ein C-Programm, das vom Benutzer 10 ganze Zahlen einliest. Erzeugen Sie zwei Felder der Größe 10 (für bis zu 10 Integer-Werte): Das Feld `int odds[10];` soll ungerade Zahlen speichern, das Feld `int evens[10];` soll gerade Zahlen speichern. Ihr Programm soll bei jeder eingegebenen Zahl überprüfen, ob eine gerade oder eine ungerade Zahl eingegeben wurde, und diese Zahl entsprechend dem Feld `odds` oder dem Feld `evens` hinzugefügt werden. Überlegen Sie, wie Sie sich merken können, wie viele gerade/ungerade Zahlen Sie bereits in das jeweilige Feld eingetragen haben. Geben Sie abschließend die Inhalte der beiden Felder vollständig aus, so dass zunächst alle ungeraden eingegebenen Zahlen und anschließend alle gerade eingegebenen Zahlen ausgegeben werden.

## Aufgabe 43: Wortlänge

### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Länge einer Zeichenkette
- Schleifen

### Aufgabenstellung

Schreiben Sie ein Programm `wortlaenge.c`, das vom Benutzer ein Wort beliebiger Länge bekommt und in ein Feld vom Typ `char` abspeichert. Dabei können Sie wie üblich die Funktion `scanf()` benutzen.

- a) Ermitteln Sie mit Hilfe von einer `for`- bzw. einer `while`-Schleife die Anzahl der Buchstaben des Wortes.
- b) Ist das die tatsächliche Anzahl von Elementen in einer C-Zeichenkette?

### Aufgabe 44: Zeichensuche

#### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Felder als Zeichenketten
- Sonderzeichen

### Aufgabenstellung

Gegeben ist das folgende char-Feld:

```
char word[] = "asdako234kfao874f<dsj111lasd9as8kg%\"+asd90101jgj";
```

Schreiben Sie ein Programm `zeichensuche.c`, das diese Zeichenkette nach bestimmten Zeichen durchsucht und die Position von jedem Auftauchen angibt. Fordern Sie hierzu den Benutzer des Programmes auf, das zu suchende Zeichen einzugeben. Wurde das entsprechende Zeichen nicht gefunden, soll dies dem Benutzer mitgeteilt werden. Wurde es gefunden, so soll angegeben werden, an welcher Stelle bzw. an welchen Stellen es gefunden wurde. Testen Sie ihren Code z.B. mit: Anführungszeichen, Backslash, %, A, 1, o etc.

### Aufgabe 45: Ihr zufällig ausgegebener Name

#### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Felder und Zeichenketten
- Zufallsgenerator

### Aufgabenstellung

In dieser Aufgabe greifen Sie Ihre Lösung zur Aufgabe “Zufallstext” (sh. Aufgabenblatt 6) auf und erweitern diese. Zur Erinnerung: In jener Aufgabe wurde rein zufälliger Text (endlos) unter Verwendung aller druckbaren ASCII-Zeichen auf den Bildschirm ausgegeben.

Da der ausgegebene Text rein zufällig ist, ist anzunehmen, dass irgendwann in dieser endlosen zufälligen Ausgabe auch Ihr Name auftaucht.

Schreiben Sie ein C-Programm (bzw. erweitern Sie Ihr Programm zur Aufgabe “Zufallstext”), das den zufälligen Text solange ausgibt, bis erstmalig Ihr Name in der Ausgabe auftaucht!

Beherzigen Sie die folgenden Hinweise bei Ihrer Implementierung:

- Testen Sie Ihre Lösung zunächst mit einer kürzeren Zeichenkette anstatt mit Ihrem Namen (also bspw. nur mit Ihren Initialien); so werden Sie schneller zum Programmende gelangen.
- Die druckbaren ASCII-Zeichen liegen im Bereich 32 bis 127.
- Initialisieren Sie den Zufallsgenerator mit einem festen (also nicht systemzeitabhängigen) Seed (z.B. mit `srand(42);`).
- Im Erfolgsfall soll Ihr Programm auch noch ausgeben, nach wie vielen zufällig ausgegebenen Zeichen der gesuchte (Ihr) Name aufgetreten ist.
- Wann wird zufällig  $\pi$  (3.14) ausgegeben?
- Wann wird zufällig “Xmas” ausgegeben?

## Aufgabe 46: Weihnachtsbaum

### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Schleifen, ASCII-Art
- Zufallsgenerator
- weihnachtliche Stimmung ;-)

### Aufgabenstellung

Entwerfen Sie ein C-Programm `weihnachtsbaum.c`, das Ihnen einen Weihnachtsbaum auf den Bildschirm zeichnet. Das Programm soll den folgenden Anforderungen genügen (alle Breitenangaben beziehen sich auf Zeichen, alle Höhenangaben auf Zeilen in der Ausgabe):

- Der Baum soll aus einem Baumteil (Nadelteil) und einem Stamm bestehen.
- Die Mindesthöhe des Baumes (ohne Stamm) soll 3 betragen.
- Der Benutzer soll die gewünschte Gesamthöhe des Baumes eingeben können.
- Der Benutzer soll die Höhe des Stammes eingeben können.

- Das Programm soll abgebrochen werden, wenn die eingegebene Gesamthöhe kleiner ist als die Summe aus der Stammhöhe und der Baummindesthöhe.
  - Die Breite des Baumes soll doppelt so groß sein wie die Differenz aus der Gesamthöhe und der Stammhöhe.
  - Der Stamm soll 3 Zeichen breit und soll zentriert unterhalb des Nadelteils angebracht sein.
  - Der Baumteil (Nadelteil) soll symmetrisch sein (also ein gleichschenkeliges Dreieck bilden), d.h. an seiner Spitze soll der Baum eine Breite von eins haben.
  - Pro folgender Zeile soll der Nadelteil um zwei Zeichen breiter sein als in der vorangegangenen Zeile.
  - Für die Darstellung des Nadelteils und Stammteils ist das Zeichen '\*' zu benutzen.
- a) Schreiben Sie das Weihnachtsbaumprogramm, so dass es die oben aufgelisteten Anforderungen erfüllt.
  - b) Erweitern Sie Ihr Programm so, dass an Ihrem Baum *zufällig* Weihnachtskugeln platziert werden.
  - c) Stellen Sie den Baum farbig dar (grüne Nadeln, brauner Stamm, stilvolle Farbe(n) für die Kugeln, Lametta in passender Farbe, ggf. blinkende Verzierungen etc.), indem Sie kreativen Gebrauch machen von der Bibliothek *Advanced Console Output*, deren Verwendung Sie im Rahmen der Bearbeitung des vorangegangenen Aufgabenblattes geübt haben.
  - d) *Freiwillige Zusatzaufgabe:* Senden Sie Ihre Lösung zum Teil c) bis spätestens zum Vortag Ihrer Übungsstunde in Kalendwerwoche 51 an `{tgabel|dominic.gibietz}@fra-uas.de@fra-uas.de`. Im Rahmen der Übungsstunde vor Weihnachten stellen Sie Ihre Lösung vor den Teilnehmern der Übungsgruppe vor (am Dozentenrechner und Beamer). Der Autor des am ästhetischsten<sup>1</sup> aussehenden Weihnachtsbaums wird mit einer zusätzlich gutgeschriebenen Anwesenheitswoche belohnt.

Orientieren Sie sich bei der Erstellung Ihrer Lösungen an den Beispielen, die auf der folgenden Seite gegeben sind!

---

<sup>1</sup>Der Gewinner wird in der Übungsstunde mittels einer Abstimmung unter allen anwesenden Übungsgruppenteilnehmern bestimmt.

Beispiele für die Ein- und Ausgabe in (a):

Bitte die gewünschte Baumgesamthöhe eingeben: 18

Bitte die Höhe des Stamms eingeben: 2

[illegible]

Beispiele für die Ein- und Ausgabe in (b):

Bitte die gewünschte Baumgesamthöhe eingeben: 20

Bitte die Höhe des Stamms eingeben: 3

```

*
***
*****
***O***
O*****
*****
*****
O***O***O*****
*****O*
*****O*****
*O*****O*****O
*****
**O***O*****O*O*
*****
*****O*****O*O***
O*****O**O***O**
O*****O*****O*O*****
***
***
***

```

Frohe Weihnachten  
und einen guten Start ins neue Jahr!