



Praktikum Informatik II – BMT

Programmierung in C

Praktikum 6 – Arrays und Zeiger

Prof. Dr. Dieter Koller

Aufgabe 1: Funktion – Zeichen zählen

Schreiben Sie ein Programm in C, mit dem Sie zunächst einen **ganzen Satz** eingeben können. Mit einer Funktion, die Sie schreiben:

```
int stringLength(char text[])
```

prüfen Sie anschließend, wie viele Zeichen eingegeben wurden und geben den Wert aus. Benutzen Sie eine Schleife mit Zähler und beenden Sie diese, wenn das für eine Zeichenkette erforderliche Endesymbol '`\0`' gefunden wird.

Hinweis: Ganze Sätze inklusive **Leerzeichen** können nur mit `gets()` eingegeben werden. Zuweisungen einer Zeichenkette mit `scanf()` brechen beim nächsten Leerraumzeichen (Whitespace), also auch bei einem Leerzeichen, ab.

Aufgabe 2: Lottozahlen

Schreiben Sie ein Programm, das die Ziehung der Lottozahlen (6 aus 49) simuliert. Verwenden Sie dazu den Pseudo-Zufallszahlengenerator `rand()` in C, den Sie zuvor mit Hilfe der `srand()` Funktion geeignet initialisieren, und legen Sie die Lottozahlen in einem Array ab. Gehen Sie wie folgt vor und erweitern Sie schrittweise Ihr Programm:

- Geben Sie die erzeugten 6 Zufallszahlen zunächst einfach aus (z.B. durch Klammern getrennt), um diese zu überprüfen.
- Berücksichtigen Sie nun, dass beim Ziehen einer Zahl, diese im Folgenden nicht mehr gezogen werden kann (die Kugel wurde entnommen).
- Geben Sie nun die Lottozahlen der Größe nach sortiert aus. Benutzen Sie dazu den Bubblesort Algorithmus.
- Wiederholen Sie nun Ziehungen von Lottozahlen, bis auf Anfrage nach jeder Ziehung die Taste 'q' gedrückt wird. (Benutzen Sie dazu die Funktion `getch()` aus `<conio.h>` in einer `do-while()` Schleife).

```
Die Lottozahlen sind: (23) (24) (26) (31) (41) (46)
```

Aufgabe 3: char Arrays und Stringverarbeitung

Schreiben Sie ein Programm, das das char-Array:

"Drei Chinesen mit dem Kontrabass"

nach den Vokalen *"aeiou"* durchsucht und dann mit einem vorher eingegebenen Vokal (oder belieb. Zeichen) ersetzt. Das geänderte Array soll dann ausgegeben.

Welche Möglichkeiten gibt es, nur ein Zeichen einzulesen?

```
"Drei Chinesen mit dem Kontrabass"
```

```
Durch welches Zeichen sollen die Vokale ersetzt werden: i  
Ergebnis der Ersetzung:
```

```
"Drii Chinisin mit dim Kintribiss"
```

Aufgabe 4: Primzahlen

Schreiben Sie ein C-Programm, das die Primzahlen kleiner 1000 und ihre Anzahl ausgibt. Eine ganze Zahl ist eine Primzahl, wenn sie größer als 1 und nur durch 1 und sich selber teilbar ist. Die ersten Primzahlen sind also 2, 3, 5, 7, 11, ...

Verwenden Sie zur Berechnung den *Sieb des Eratosthenes*:

Es bleiben nur Primzahlen übrig, wenn man alle Vielfachen einer schon gefundenen Primzahl „streicht“, das heißt, man streicht

Alle Vielfachen von 2, also 4, 6, 8, ...

dann alle Vielfachen von 3, also 6, 9, 12, ...

dann alle Vielfachen von 5, also 10, 15, 20, ... (die 4 wurde schon gestrichen von der 2) etc.

Hinweis: Definieren Sie ein char-Array der Länge 1000, in dem zunächst alle Elemente ab dem Index 2 auf 1 (true) gesetzt sind. Die Zahl i „streichen“ bedeutet dann, das i-te Element auf 0 (false) zu setzen. Die Indizes i, die am Ende noch auf 1 stehen, sind dann Primzahlen.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	998	999
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2:	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3:	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0
5:	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0

Aufgabe 5: Funktionsminimum suchen

Schreiben Sie ein Programm, bei dem zunächst die obere und untere Grenze eines Intervalls als Fließkommazahlen eingeben können. Dann füllen Sie ein x-Array mit 11 Werten gleichmäßig zwischen diesen Grenzen (von der unteren Grenze $x[0]$ bis zur oberen Grenze $x[10]$).

Geben Sie diese Werte dann in einer Zeile mit drei Nachkommastellen in einem 8 Zeichen breiten Feld rechtsbündig aus (siehe unten).

Berechnen Sie dann für diese 11 x-Werte auch 11 y-Werte in einem Array mit $y = 2x^2 - 5x + 1$ und bestimmen Sie das Minimum der y-Werte. Geben Sie dann die y-Werte in der Zeile unter den x-Werten (wieder mit 3 Nachkommastellen in einem 8 Zeichen breiten Feld) und das Minimum aus.

Wie muss das Programm erweitert werden, um nach einen weiteren Durchlauf mit 10 neuen Grenzen eine größere Genauigkeit zu erzielen? Dabei soll die neue Untergrenze der linke Nachbarwert und die neue Obergrenze der rechte Nachbarwert des vorherigen Minimums sein.

```
Untere Grenze: -5
Obere Grenze: 5
x =   -5.000  -4.000  -3.000  -2.000  -1.000   0.000   1.000   2.000   3.000   4.000   5.000
y =   76.000  53.000  34.000  19.000   8.000   1.000  -2.000  -1.000   4.000  13.000  26.000
Das Minimum bei x = 1.000 ist ymin = -2.000

Neues Intervall: [0.000 ... 2.000]
x =    0.000   0.200   0.400   0.600   0.800   1.000   1.200   1.400   1.600   1.800   2.000
y =    1.000   0.080  -0.680  -1.280  -1.720  -2.000  -2.120  -2.080  -1.880  -1.520  -1.000
Das Minimum bei x = 1.200 ist ymin = -2.120
```