

Wintersemester 2005/2006

Übungen zu Grundlagen der Programmierung in C - Blatt VI

Abgabe vom 14.12.2005 bis 20.12.2005 in der angemeldeten Übung

Aufgabe 1 (Array Sortierung, 5 Punkte)

Hinweise:

Durchlaufe das Array immer wieder und vertausche wenn nötig benachbarte Elemente.
Ist in einem Durchlauf kein Austausch mehr erforderlich, ist das Array sortiert.

1. Schreiben Sie ein Programm, das 10 `integer` Zahlen in ein `array` einliest und es in aufsteigender Reihenfolge sortiert. Geben Sie danach das sortierte Feld mit der größten Zahl beginnend aus.

Aufgabe 2 (Sieb des Eratosthenes, 8 Punkte)

Hinweise:

Das Sieb des Eratosthenes beschreibt ein seit über 2000 Jahren bekanntes einfaches Verfahren, um alle Primzahlen im Bereich von 1 bis zu einem bestimmten Wert `n` zu ermitteln. Man legt ein `n`-Elemente großes Array an und initialisiert es mit `wahr`, was bedeutet die zum Element gehörige Zahl könnte eine Primzahl sein. Danach durchläuft man das Array und setzt alle Produkte (nicht Primzahlen) auf `falsch` und kennzeichnet sie damit als nicht Primzahlen.

Das Verfahren läßt sich folgendermaßen beschreiben:

- Erstellen Sie ein boole'sches Array mit `n` Elementen. Den Elementen sind die Zahlen 1 bis `n` zugeordnet. Initialisieren Sie alle Elemente mit `true`.
- Ändern Sie das erste Element mit Index 0, dem die 1 zugeordnet ist, auf `false`, da 1 keine Primzahl ist.
- Nun führen Sie in einer Schleife folgende Schritte aus:
 - Suchen Sie das nächste Element, dessen Wert `true` ist und setzen Sie alle Produkte größer der Zahl selbst, in denen diese Zahl vorkommt auf `false`.
Es beginnt bei 2 (Index 1) mit : $2 * 2, 2 * 3, \dots, 2 * x$, man kann aufhören, wenn $2 * x > n$ ist.
Dann folgt 3 (Index 2) mit : $3 * 2, 3 * 3, \dots, 3 * x$, wobei $3 * x > n$ ist.
5 ... 7 ...
Falls kein weiteres Element diese Bedingung (`true`) erfüllt, beenden Sie die Schleife.
 - Geben Sie nun alle Zahlen aus, die `array`-Elementen zugeordnet sind, deren Wert `true` ist.
Es sind die Primzahlen im Bereich von 1 bis `n`.

1. Schreiben Sie ein Programm, das das "Sieb des Eratosthenes" realisiert und geben Sie auf diese Weise alle Primzahlen zwischen 1 und 1000 aus.

Aufgabe 3 (Determinante, 6 Punkte)

1. Schreiben Sie ein Programm, das eine 3 x 3 Matrix einliest, diese in einem 2-dimensionalen `array` speichert und die zugehörige Determinante berechnet. Anschließend soll sowohl die Matrix als auch ihre Determinante, ähnlich wie in diesem Beispiel, ausgegeben werden:

```
Matrix A =
```

```
 2 12  5
11  7  9
 6  4  3
```

```
det(A) = 232
```

Aufgabe 4 (Stringvergleich, 4 Punkte)

Hinweise:

Die Strings sollen maximal 60 Zeichen lang sein.

Benutzen Sie auch hier wieder `gets()` zum Einlesen von Strings.

Ignorieren Sie ebenfalls wieder die Warnung des Compilers zu `gets`, falls dieser eine ausgibt.

Alle anderen Warnungen sollen jedoch beachtet werden.

1. Schreiben Sie ein Programm, das zwei Strings einliest und diese zeichenweise auf Gleichheit überprüft. Geben Sie das Ergebnis des Vergleichs (gleich oder ungleich) aus.
Achtung: Die Aufgabe soll ohne Verwendung der in `string.h` deklarierten Funktionen gelöst werden!